



Traitement générique des interactions haptiques pour l'assemblage d'objets issus de CAO

Loïc Tching

► To cite this version:

Loïc Tching. Traitement générique des interactions haptiques pour l'assemblage d'objets issus de CAO. Autre [cs.OH]. INSA de Rennes, 2010. Français. NNT : . tel-00470327

HAL Id: tel-00470327

<https://theses.hal.science/tel-00470327>

Submitted on 6 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: D10-01

THÈSE

Présentée devant

devant l'INSA de Rennes

pour obtenir

le grade de : DOCTEUR DE L'INSTITUT NATIONAL DES SCIENCES
APPLIQUÉES DE RENNES
Mention INFORMATIQUE

par

Loïc TCHING

Équipe d'accueil : UMR IRISA - EPI Bunraku

École Doctorale : Matisse

Composante universitaire : UNIVERSITÉ EUROPÉENNE DE BRETAGNE

Titre de la thèse :

*Traitement générique des interactions haptiques pour
l'assemblage d'objets issus de CAO*

soutenue le 4 février 2010 devant la commission d'examen

M. Bruno Arnaldi	Professeur de l'INSA de Rennes	Président du jury
M. Jean-Claude Léon	Professeur de l'ENSE3 Grenoble	Rapporteur
M. Xavier Fischer	Professeur de l'ESTIA, HDR	Rapporteur
Mme. Judy Vance	Professeur de l'Iowa State University	Examinatrice
M. Georges Dumont	Maître de Conférences de l'ENS Cachan, HDR	Directeur de thèse
M. Jérôme Perret	Directeur Général de Haption SA, Docteur	Encadrement
M. Claude Andriot	Docteur, Ingénieur de Recherche du CEA-LIST	Invité

Dans l'échange entre la théorie et l'expérience, c'est toujours la première qui engage le dialogue. C'est elle qui détermine la forme de la question, donc les limites de la réponse.

par François Jacob

Remerciements

Je tiens en tout premier lieu à remercier mes directeurs de thèse, Georges Dumont et Jérôme Perret, sans qui toute cette aventure n'aurait pas eu lieu. J'ai pour ces deux personnes un profond respect et une grande reconnaissance. Parmi les mille et un points sur lesquels je voudrais les remercier, je retiendrais leur confiance, leur disponibilité et les conseils qu'ils m'ont prodigués sans relâche pendant ces trois années.

Je remercie naturellement les membres de mon jury. Je remercie le Professeur Bruno Arnaldi qui me fait l'honneur de présider ce jury et auquel j'adresse, à titre personnel, mon plus profond respect. Je remercie le Professeur Jean-Claude Léon, et le Professeur Xavier Fischer, d'avoir bien voulu accepter la charge de rapporteur. Je remercie le Professeur Judy Vance et Monsieur Claude Andriot d'avoir bien voulu juger ce travail.

Je tiens à remercier l'ensemble des personnes qui ont contribué à l'élaboration de ce travail. Je remercie Bernard Brogliato et Vincent Acary de l'équipe Bipop pour leur accueil et leur collaboration. Merci à Antonio Capobianco et Caroline Essert du LSIIT pour leurs conseils avisés. Merci aux membres du CEA LIST et particulièrement à Jean Sreng. J'adresse également mes remerciements à l'équipe d'Haption pour son soutien.

Je n'oublie pas les ex-siamois, golaemiens et zukaris dont la liste serait trop longue à écrire. Beaucoup d'entre eux ont joué un rôle important dans cette thèse. J'ai rencontré parmi eux des gens passionnés, qui m'ont donné le goût de la recherche. J'ai également rencontré des professionnels incroyables, des gens généreux et patients qui m'ont transmis les ficelles du métier. Grâce à eux j'ai surtout découvert des amis qui m'ont permis de vivre trois merveilleuses années.

Je remercie enfin Angélique mon aimée, ma famille et mes amis.

Table des matières

Remerciements	1
Table des matières	2
Introduction	7
1 Interactions haptiques dans le cadre de la réalité virtuelle	11
1.1 Introduction	11
1.2 Enjeux de la RV pour la CAO	12
1.3 Interactions haptiques : problématiques	13
1.3.1 Notions de qualité de rendu	13
1.3.2 Architecture des interfaces haptiques	14
1.3.3 Couplage entre l'interface et le simulateur	16
1.4 Dissociation d'une tâche d'assemblage interactive	19
1.5 Conclusion sur l'état de l'art de l'haptique généralisée	20
2 Dynamique non régulière appliquée à l'haptique	23
2.1 Introduction	23
2.2 Etat de l'art du traitement des interactions dans les simulations haptiques	24
2.2.1 Détection des collisions	24
2.2.2 Détermination des efforts d'interaction	27
2.2.3 Modèles pénalisés	28
2.2.4 Exemples industriels basés sur le modèle pénalisé	29
2.2.5 Lois de contact	31
2.2.6 Modèles non réguliers (<i>non-smooth dynamics</i>)	34
2.2.7 Approche par événements (<i>event-driven</i>)	34
2.2.8 Approche temps-contrôlé (<i>time-stepping</i>)	36
2.2.9 Conclusion sur l'état de l'art des modèles de traitement des interactions haptiques	37
2.3 Contribution : intégration de la NSCD dans une simulation haptique	37

2.3.1	Conf&tis : plate-forme de simulation haptique par dynamique non régulière	38
2.3.2	Couplage avec une interface à retour d'effort	41
2.4	Conclusion	42
3	Guidage virtuel contraint pour l'assemblage d'objets CAO	43
3.1	Introduction	43
3.2	Etat de l'art des assemblages d'objets CAO en environnement RV	45
3.2.1	Exploitation des objets issus de CAO	45
3.2.2	Limites de la détection des collisions dans les simulations d'assemblage	51
3.2.3	La métaphore des guides virtuels	55
3.2.4	Conclusion sur les assemblages d'objets CAO en environnement virtuel	57
3.3	Contribution : guidage virtuel contraint pour l'assemblage d'objets CAO	58
3.3.1	Concept : considérer l'assemblage comme une combinaison de liaisons mécaniques	59
3.3.2	Guidage géométrique d'objets 3D par entités virtuelles conjuguées	71
3.3.3	Description des géométries de guidage	72
3.3.4	Interaction entre les géométries maillées des objets et les guides géométriques 3D	75
3.3.5	Interaction entre les représentations abstraites des objets et des guides géométriques 3D	77
3.3.6	Propositions sur l'utilisation des guides géométriques . . .	80
3.3.7	Changement de mode de contrôle : activation du guidage contraint	82
3.4	Conclusion	88
4	Expériences et démonstrations de la phase exploratoire et de la phase d'assemblage	91
4.1	Introduction	91
4.2	Exploration haptique de scènes virtuelles par dynamique non régulière	93
4.2.1	Expériences et résultats	93
4.2.2	Conclusion sur l'exploration des scènes haptiques par dynamique non régulière	97
4.3	Assemblages haptiques d'objets CAO par guidage virtuel contraint	99
4.3.1	Objectif des expérimentations	99
4.3.2	Plate-forme technologique	100

4.3.3	Expérience préliminaire sur le guidage contraint	102
4.3.4	Choix de l'expérimentation	103
4.3.5	Description et spécificités des essais	106
4.3.6	Description d'un essai et des informations recueillies	110
4.3.7	Construction du plan d'analyse expérimentale	112
4.3.8	Analyse des données expérimentales pour la performance .	113
4.3.9	Performance temporelle pour chaque cas de guidage	116
4.3.10	Analyse de l'influence de la localisation des cibles sur les performances	116
4.3.11	Bilan des expériences	124
4.4	Démonstration	125
4.5	Conclusion	127
Conclusion		129
Bibliographie		145
Table des figures		147

Introduction

L'industrie a vu émerger, dès les années 80, une nouvelle forme d'organisation qui est l'ingénierie simultanée, aussi appelée *concurrent engineering*. Cette approche consiste en une mise en parallèle des activités de développement du produit et du système de production, dans le but de réduire les coûts et les délais d'industrialisation [JPS93]. En parallélisant les actions des différents intervenants du développement d'un produit, cette méthode permet d'intégrer les spécifications de chaque métier intervenant dans le cycle de vie du produit. L'approche de *conception intégrée* précise cette méthode d'ingénierie simultanée en s'appuyant sur l'échange dynamique des connaissances entre les acteurs de chaque profession, afin de valider rapidement les décisions. Les logiciels de conception assistée par ordinateur (CAO) sont ainsi dotés de systèmes d'aide au travail collaboratif reliant les services de l'entreprise entre eux et avec leurs partenaires externes. Pour le développement rapide et à moindre coût d'un produit, il est devenu primordial de considérer les différentes contraintes liées à l'industrialisation d'un produit dès les phases amonts de la conception.

L'utilisation d'outils tels que les systèmes de réalité virtuelle (RV) donne une nouvelle dimension en termes de développement efficace du produit, en permettant de travailler de manière collaborative autour de la *maquette numérique*. Parmi ces outils figurent les interfaces haptiques, qui font interagir les utilisateurs avec des simulations *temps-réel*. Elles permettent à l'utilisateur de sentir les contacts et de les utiliser comme guides lors de la manipulation d'objets 3D. En rendant compte kinesthésiquement des interactions entre les objets, l'haptique permet de recréer une interaction physique réelle avec les maquettes numériques ; ceci est fondamental pour appréhender les volumes et les contraintes dans les assemblages interactifs d'objets.

L'association de la CAO et de la RV a été tentée dès l'émergence de la réalité virtuelle. Mais, cette alliance a longtemps consisté à concevoir des objets en CAO, à les transposer en RV pour les tester, et à revenir à la CAO pour corriger les défauts identifiés. Aujourd'hui il existe de nombreuses solutions qui associent, dans un outil commun, la CAO et la RV pour simuler les assemblages collaboratifs d'objets 3D.

Les simulations d'objets 3D permettent de modéliser des comportements dy-

namiques complexes, le plus fidèlement possible à la réalité. Ces comportements sont déterminés par des *moteurs physiques*. Depuis le début des années 2000, l’explosion des moteurs physiques impose une concurrence acharnée entre les fabricants de matériel (NVidia, Intel, etc.). Portés par les domaines des jeux vidéos et du cinéma, ces moteurs sont essentiellement utilisés pour calculer les mouvements des objets. En matière de simulation physique, les domaines de l’aide à l’industrialisation et de l’interaction en environnements virtuels ont également vu leurs exigences émerger. Toutefois, les moteurs physiques précédents n’offrent qu’un réalisme limité, avec des scènes peu complexes (surtout composées de cubes), qui sont insuffisants pour le domaine de l’industrie.

La simulation d’assemblage CAO en réalité virtuelle suppose que les modèles mécaniques utilisés soient conformes à la réalité, dans la mesure où les résultats de la simulation doivent être très proches du comportement des objets matériels réels. C’est le moteur physique qui a pour tâche de calculer les comportements dynamiques des objets qui composent la scène virtuelle ; ces comportements peuvent être calculés par simulation dynamique (ou statique) ou bien par des procédés cinématiques. La modélisation et l’utilisation d’objets rigides, c’est-à-dire des objets virtuels dont on fait l’hypothèse qu’ils possèdent une rigidité infinie, est très avantageuse et largement présente dans les simulations d’assemblage. En effet, la configuration d’un objet peut alors être décrite de manière très concise sous la forme d’une position (un vecteur) et d’une orientation (un quaternion ou une matrice). En outre, cette simplicité de représentation permet à la simulation physique de mettre à jour rapidement l’état physique des objets, et ce suffisamment pour qu’un utilisateur puisse interagir en temps-réel avec la simulation d’assemblage.

Dans ce contexte, nous proposons d’appliquer les technologies de l’interaction haptique, pour la réalisation d’assemblages mécaniques issus de la CAO. Le principal enjeu de la simulation interactive d’assemblage d’objets rigides tient dans la fidélité de la simulation, par rapport à la réalité. Parmi les nombreux facteurs qui influencent le degré de réalisme de la simulation, nous nous concentrons sur :

- le réalisme des comportements physiques des objets 3D simulés ;
- les performances de l’utilisateur dans la réalisation de tâches d’assemblage.

Dans le cadre précis des simulations haptiques, nous proposons de traiter simultanément ces deux facteurs. Au sein d’une même simulation d’assemblage, l’utilisateur doit pouvoir percevoir des comportements dynamiques réalistes lors d’interactions élémentaires avec l’environnement virtuel, ainsi que pouvoir réaliser des tâches d’assemblage de manière efficace. Cette thèse propose ainsi d’utiliser deux formulations différentes de la dynamique, selon que l’utilisateur a besoin de se concentrer sur l’exploration haptique d’une scène ou sur l’assemblage d’objets CAO. Pour cela, nous proposons de baser la simulation physique sur une modélisation rigoureuse de la mécanique. Pour rendre compte de comportements réalistes des objets, nous utilisons les méthodes de dynamique non régulière, et

particulièrement des approches de *non smooth contact dynamics* (NSCD). En identifiant les différentes étapes qui composent l'assemblage interactif, nous proposons également d'assister l'utilisateur pour la réalisation de tâches de montage et de démontage. Cette assistance consiste en un guidage virtuel des objets, qui contraint géométriquement et cinématiquement les mouvements des objets à assembler.

Dans un premier chapitre, nous décrivons le processus d'interaction haptique dans le cadre de la réalité virtuelle. Dans un premier temps, nous décrivons les principes de la simulation interactive, et des liens qu'elle possède avec les assemblages CAO. Nous décrivons alors les principes de mise en œuvre du retour d'effort ainsi que les solutions existantes gérant les simulations haptiques.

Dans un second chapitre, nous présentons un état de l'art sur le traitement des contacts et des chocs au sein de simulations dynamiques. Nous soulignons pour cela les dépendances qui existent entre les méthodes de détection des collisions et les méthodes de détermination des efforts d'interaction pour leur utilisation dans des applications haptiques. Nous présentons alors nos travaux sur l'intégration haptique de la méthode *time-stepping* de la dynamique non régulière, à travers une plateforme expérimentale nommée Conf&is.

Dans un troisième chapitre, nous présentons une nouvelle méthode d'assistance à l'assemblage d'objets CAO en environnement virtuel. En combinant l'utilisation de contraintes cinématiques et de *virtual fixtures* dans la simulation dynamique, cette méthode propose deux principales contributions. La première propose une nouvelle technique de guidage géométrique pour assister la mise en position relative des objets à assembler. La seconde porte sur le changement de mode de contrôle haptique pour contraindre ces objets à réaliser l'assemblage final.

Dans un quatrième chapitre, nous présentons, en deux temps, nos expérimentations sur l'exploration haptique de scènes virtuelles par dynamique non régulière et sur notre méthode de guidage virtuel par dynamique contrainte. Dans un premier temps, nous montrons la faisabilité du couplage entre la dynamique non régulière et l'interaction haptique, puis nous mesurons les performances algorithmiques de notre méthode de couplage pour la manipulation d'objets simples. Dans un deuxième temps, nous expérimentons l'utilisabilité et les performances temporelles offertes par notre méthode de guidage, sur l'exemple canonique du *peg-in-hole*.

Nous concluons, dans un dernier chapitre, sur l'utilisation simultanée des deux modes de contrôle dynamique (contraint et non contraint) pour des assemblages CAO. Nous basons nos réflexions sur les résultats temporels de notre simulateur à dynamique non régulière, et sur le gain en performance proposé par notre méthode de guidage virtuel. Nous proposons enfin, en perspectives, des pistes d'amélioration de notre méthode, pour la manipulation et l'assemblage d'objets

CAO complexes.

Cette thèse est réalisée dans le cadre d'une convention CIFRE entre la société HAPTION et l'INSA de Rennes. Le travail que nous présentons dans ce mémoire s'inscrit dans le cadre du projet RNTL-Part@ge dont la vocation est de lever des verrous scientifiques et techniques concernant l'interaction collaborative en environnement 3D.

Chapitre 1

Interactions haptiques dans le cadre de la réalité virtuelle

1.1 Introduction

Les interfaces à retour d'effort, dites haptiques, fournissent des informations aux utilisateurs sur l'état d'objets virtuels pilotés au sein de simulations multimodales. Ces informations sont perçues par le sens du toucher et représentent les efforts de réaction qui s'exercent sur ces objets dans l'environnement.

Historiquement issue de la télé-robotique en milieu nucléaire dans les années 50, l'haptique est aujourd'hui utilisée couramment, dans beaucoup de secteurs industriels pour des applications aussi diversifiées que la formation, la validation industrielle ou l'ergonomie. Le domaine de la réalité virtuelle profite, quant à lui, des performances de calcul croissantes pour intégrer les modalités haptiques dans le cadre de simulations multimodales complexes [FMT⁺03]. Du point de vue de l'utilisateur, les enjeux portent principalement sur la rapidité et la fidélité de la simulation afin de répondre aux critères de performance et de confort d'utilisation, ainsi que proposés par Ernst et Banks [EB02]. Ces deux critères sont toutefois remis en cause lorsque les objets simulés doivent adopter des comportements difficiles à modéliser, tels que les contacts et les chocs. En effet, la présence d'interactions (de contact ou de choc) entre les objets impose la modélisation des discontinuités liées à ces interactions (caractère unilatéral des contacts et des chocs) et des lois de comportement, en particulier dans les cas où il est nécessaire de simuler le frottement. Pour des simulations traitant de nombreux contacts simultanés, l'enjeu réside donc en la représentation réaliste de ces interactions entre objets.

De nombreuses solutions, issues des domaines de la téléopération et de la réalité virtuelle existent pour traiter efficacement ces problèmes. L'approche mise en œuvre consiste généralement à utiliser des modèles mécaniques, c'est-à-dire

en dotant le monde virtuel de propriétés et de lois physiques semblables à celles qui régissent le monde réel. Les interfaces haptiques ont longtemps été utilisées dans des simulations pour effectuer des tâches purement géométriques en boucle ouverte telles que le suivi de surfaces ou dans des simulations utilisant des modélisations de mécanique quasi statique. Afin de fournir des résultats physiques plus proches de la réalité, les simulateurs se basent à présent sur des modélisations dynamiques.

1.2 Enjeux de la RV pour la CAO

La CAO est une discipline ayant recours aux techniques informatiques pour créer un objet, en achever la forme et engendrer les données nécessaires à sa fabrication. De nombreux domaines (la mécanique, l'aéronautique, l'architecture etc.) font appel à la conception assistée par ordinateur. Les logiciels de CAO disponibles sur le marché permettent, de manière plus ou moins automatisée, de concevoir et d'analyser des pièces et des produits. Malgré le perfectionnement des outils d'analyse par éléments finis, d'ergonomie, de cinématique etc., il existe toujours un vide important entre les phases de conception et de fabrication. En effet, le processus de conception d'objets complexes nécessite, de plus en plus, la participation simultanée, d'un nombre croissant d'experts de métiers différents. Ces experts ont des perceptions différentes du même produit, ce qui définit la multi-représentation du produit. Ces vues comprennent un ensemble d'informations qui leur sont spécifiques et, notamment, un ou plusieurs sous modèles qui sont entièrement dédiés au métier considéré. Dans le domaine de l'assemblage mécanique, les logiciels CAO, tout en aidant le concepteur dans sa tâche, masquent des problèmes apparaissant uniquement lors de la phase de fabrication. Lors de la phase CAO, les concepteurs en bureau d'études sont rarement confrontés à la réalité du montage. Les difficultés de mise en position relative de deux pièces avant fixation ou encore les difficultés d'insertion d'une pièce par rapport aux autres sont autant de problèmes récurrents de l'inaccessibilité ou de la mauvaise gestion des collisions lors des phases d'assemblage d'objets.

Durant la conception assistée par ordinateur d'un produit, le processus d'assemblage est souvent négligé, voire ignoré, de part la séparation historique de la conception (CAO) de la production (FAO). Les pièces d'un assemblage sont placées directement à leurs positions finales ; ces positions sont définies par des contraintes géométriques entre les pièces de l'assemblage. La vérification ou la simulation de ces assemblages n'est réalisée qu'*a posteriori*, et uniquement en termes de trajectoires et de collisions. Les modeleurs CAO récents, ainsi que des solutions commerciales externes telle que eM-Assembler, proposent la simulation d'assemblage directement dans les logiciels de conception, mais *a posteriori*

dans le processus de conception. Les séquences et trajectoires proposées sont, par ailleurs, prédéfinies de sorte que les opérateurs ne peuvent réaliser l'assemblage que de manière intuitive et avec une interactivité limitée. Mikchevitch, dans ses travaux de thèse [Mik04], propose un état de l'art complet des simulations d'assemblage CAO en environnement virtuel.

1.3 Interactions haptiques : problématiques

1.3.1 Notions de qualité de rendu

On peut juger de la qualité d'un rendu haptique à partir du compromis établi entre la vitesse de la simulation, la transparence de l'interface et la raideur restituée par l'interface physique au moment des contacts. Afin de faire parvenir des informations immédiates à l'utilisateur et de lui donner l'impression de manipuler les objets naturellement, ces trois facteurs doivent être évalués en fonction des applications souhaitées et doivent respecter un critère de stabilité (voir section 1.3.3).

Au regard de l'immersion dans un environnement virtuel, la présence active de l'utilisateur impose une contrainte fréquentielle à la simulation. En effet, l'instantanéité souhaitée entre la commande et le retour d'information est assimilable à une contrainte temporelle de *temps-interactif*, qui impose à la vitesse de la simulation d'assurer un flux continu et temporellement maîtrisé d'informations. Pour que l'utilisateur puisse interagir de façon naturelle avec les objets virtuels, la perception subjective du taux de rafraîchissement kinesthésique se situe dans une bande de fréquence d'environ 1 kHz [Bur99]. Pour un retour haptique, ce taux est en pratique de l'ordre de 30 Hz à 300 Hz pour la plupart des applications, alors même qu'un humain peut contrôler un mouvement dans une bande maximale de 10 Hz [KFN06].

La transparence peut être définie comme l'aisance d'un utilisateur à manipuler des objets dans un environnement virtuel. Elle doit assurer à l'utilisateur que les efforts ressentis sont propres à la simulation et doit atténuer le plus possible les perturbations liées à l'interface physique [Law93]. La principale caractéristique de cette transparence est l'absence de ressenti par l'utilisateur d'éléments tels que le poids, l'inertie, les liaisons mécaniques ou les actionneurs du dispositif physique [Gos05]. La raideur, quant à elle, peut être considérée comme la dureté ressentie, au sens kinesthésique, lors d'un contact entre un objet piloté et un environnement. Pour faire état du réalisme de la raideur d'une interface, on peut distinguer la raideur mécanique définie comme sa résistance à subir une déformation sous l'action d'une force [SV97], de la raideur de son asservissement issue des lois de l'automatique telle que Colgate [CB94] la décrit pour le domaine de la téléopération.

La transmission des informations entre les différentes parties de l'interface haptique et du simulateur peut créer une latence entre l'action de l'utilisateur et la réaction. Cette latence, qui est d'autant plus importante que le système maître est éloigné du système esclave, est perçue comme une perte de raideur au niveau du rendu haptique, avec un « effet éponge molle » [NS97]. Le rendu haptique doit donc faire état des interactions entre objets de manière réaliste, en assurant un transfert rapide d'informations, indépendant de l'interface et restituant le mieux possible la raideur des phénomènes de contact.

1.3.2 Architecture des interfaces haptiques

[HACH⁺04] et [SCB04] dressent un bilan sur les applications et les architectures des interfaces haptiques. D'un point de vue global, une interface haptique peut être considérée comme un système maître-esclave régi par des lois de commande. Historiquement issue des recherches dans le domaine de la téléopération [Goe64], cette approche automatique est aujourd'hui largement appliquée aux environnements virtuels.

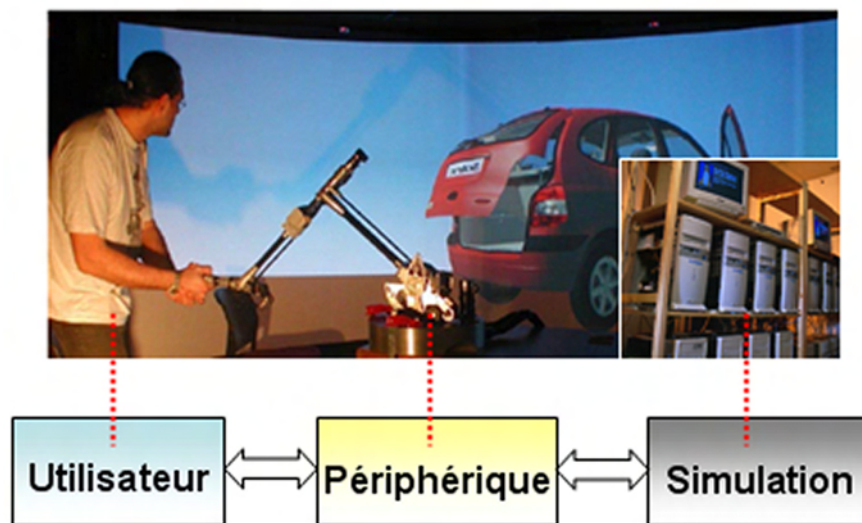


FIG. 1.1 – Interaction haptique

Le processus d'interaction entre l'utilisateur et l'environnement simulé peut se décomposer en trois processus indépendants, qui communiquent entre eux de manière synchrone ou asynchrone (voir figure 1.1) :

- dans le processus de commande, l'utilisateur actionne une interface physique dont la structure mécanique peut varier entre une chaîne cinématique ouverte [MS94], une chaîne fermée [RH94], une structure à câbles tendus [KHKS02] ou de type magnétique [BHS95]. Selon la complexité de

- l'interface, le contrôle de l'objet peut s'établir selon plusieurs degrés de liberté (ddl) ;
- dans le processus de calcul, le simulateur calcule les informations de mise à jour de la simulation en fonction des informations de commande et des lois d'évolution des modèles physiques du simulateur. L'objet piloté peut ainsi se déplacer librement dans l'environnement ou entrer en contact avec d'autres objets simulés. Le simulateur est chargé de détecter et traiter les contacts et les collisions entre les différents objets de la simulation et de calculer les réponses à transmettre à l'utilisateur en retour ;
 - dans le processus de retour d'information, l'utilisateur reçoit une information haptique, dite retour d'effort liée aux efforts déterminés dans l'environnement simulé. Le rendu haptique, qui correspond à la qualité de restitution des interactions avec les objets de la scène, est ainsi perçu par l'utilisateur. Il est toutefois courant de considérer simultanément d'autres retours sensoriels (par exemple visuels) durant cette phase, ce qui permet, par combinaison, d'induire des sensations de type pseudo-haptique [LAC03].

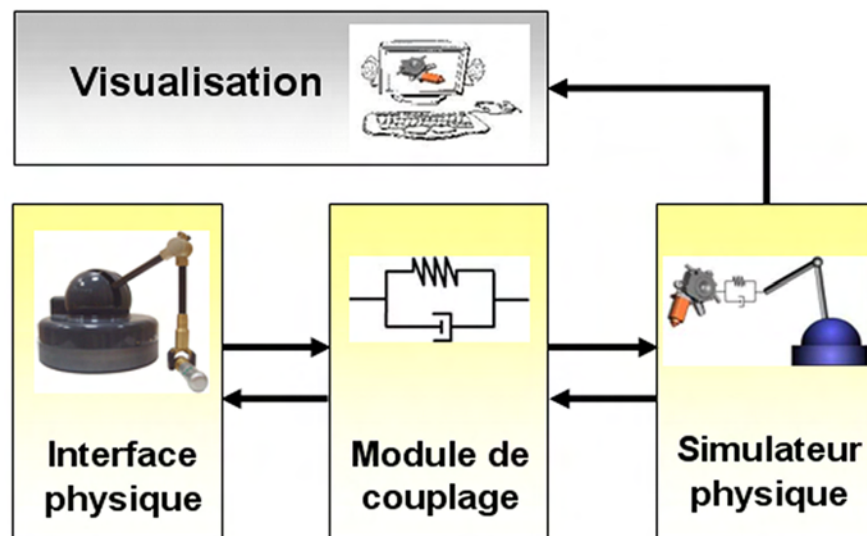


FIG. 1.2 – Schéma de principe d'un système haptique local

Nous proposons de décomposer un système à retour d'effort local en quatre modules, qui peuvent fonctionner d'une manière parallèle, par échange synchronisé de leurs données d'entrée/sortie respectives (voir figure 1.2) :

- une interface physique accompagnée d'une unité de contrôle de ses axes mécaniques ;
- un module qui assure la communication et le couplage entre le contrôleur de cette interface et le simulateur dynamique ;

- un logiciel de simulation physique temps-réel qui intègre les équations de la dynamique ainsi que les algorithmes de détection et traitement des collisions ;
- un système de visualisation qui affiche les objets calculés par le simulateur.

La plupart des systèmes haptiques opèrent à l'aide d'une boucle d'interaction fermée et restituent les efforts d'interaction à la fréquence du mouvement de l'utilisateur. Hwang *et al.* [HWN04] abandonnent cette approche en boucle fermée pour engendrer des signaux hautes fréquences en boucle ouverte. Les forces correspondant à un contact donné sont précalculées en fonction des conditions de contact (vitesse, matériaux etc.), puis superposées aux signaux basses fréquences issus de la simulation. La même idée est proposée par Kuchenbecker [KFN05] ; dans ces travaux, les forces peuvent être précalculées à partir d'observations basées sur des contacts réels entre objets constitués de matériaux donnés. Cette approche par événements impose de travailler avec des bibliothèques d'efforts qui interdisent la restitution de forces d'interaction uniquement basées sur les données calculées par la simulation en ligne. Une autre méthode, proposée par Constantinescu [CSC05], consiste à séparer la boucle de contrôle des efforts de l'environnement virtuel, par l'utilisation d'un modèle local d'interaction ; en réduisant ainsi le nombre de tâches allouées à la simulation, les scènes virtuelles peuvent alors contenir des objets plus complexes pour les mêmes performances temporelles.

1.3.3 Couplage entre l'interface et le simulateur

Les interfaces haptiques sont bidirectionnelles, c'est-à-dire qu'elles retournent explicitement des informations d'effort ou de position, à l'utilisateur. Selon son architecture, un simulateur physique :

- nécessite des variables d'effort en entrée et retourne des informations de position et de vitesse en sortie ;
- nécessite des variables de position / vitesse en entrée et retourne des efforts en sortie.

De même, les interfaces peuvent être contrôlées selon deux modes :

- en admittance et dans ce cas elles retournent des informations de position / vitesse et nécessitent des informations d'effort en entrée ;
- en impédance et dans ce cas elles retournent des informations d'effort et nécessitent des informations de position / vitesse en entrée [AH98].

Il incombe alors à un module de couplage (voir figure 1.3) de convertir et transmettre les données respectives entre l'interface haptique et le simulateur selon les différentes combinaisons possibles.

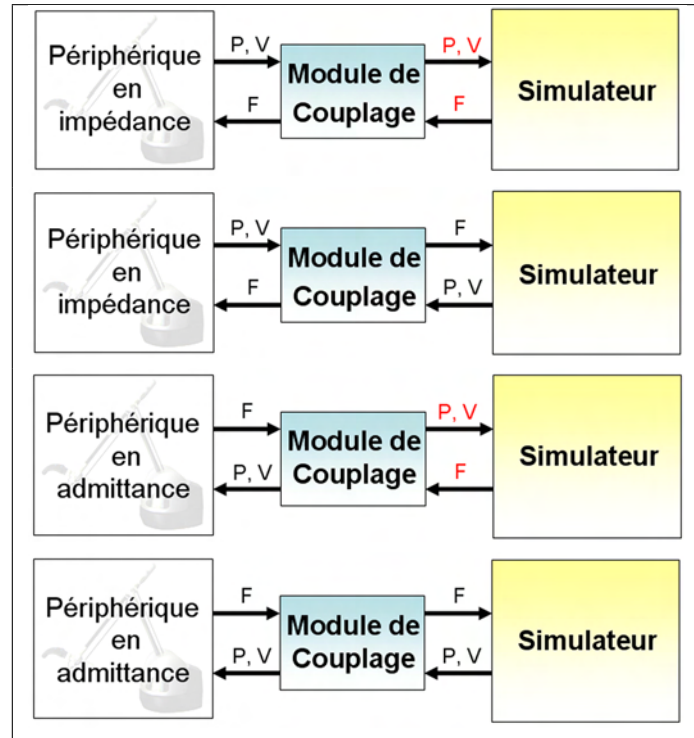


FIG. 1.3 – Différentes combinaisons de couplage entre le périphérique et le simulateur.

Ce couplage peut, par exemple, être réalisé par l'introduction d'un ressort en parallèle à un amortisseur virtuel, entre le pointeur de l'interface haptique et l'objet simulé [GC97]. Cette méthode d'indirection entre l'interface et le simulateur dynamique est appelée couplage virtuel [CSB95] (voir figure 1.4).

L'étude de la stabilité du dispositif consiste, pour la boucle de commande, à modifier les forces calculées par l'environnement de la simulation afin de les rendre compatibles avec les forces maximales délivrables par le dispositif, dans toutes les conditions d'utilisation possibles [MCF00]. En pratique, le couplage par ressort-amortisseur nécessite des réglages qui réalisent les compromis entre la stabilité et la transparence du système [LMCL01]. De même, le réglage des coefficients de raideur et d'amortissement n'est généralement pas trivial et influence directement la raideur du rendu haptique. L'augmentation du coefficient d'amortissement de l'amortisseur virtuel induit la création d'un retard entre la commande et le retour. La viscosité ainsi induite altère la qualité du rendu à l'utilisateur par un effet de mollesse.

Colgate *et al.* [CB94] examinent précisément ces problèmes de stabilité par l'élaboration de tests de référence (*benchmarks*) dont l'objectif est de déterminer les conditions sous lesquelles l'interface haptique gardera un comportement passif.

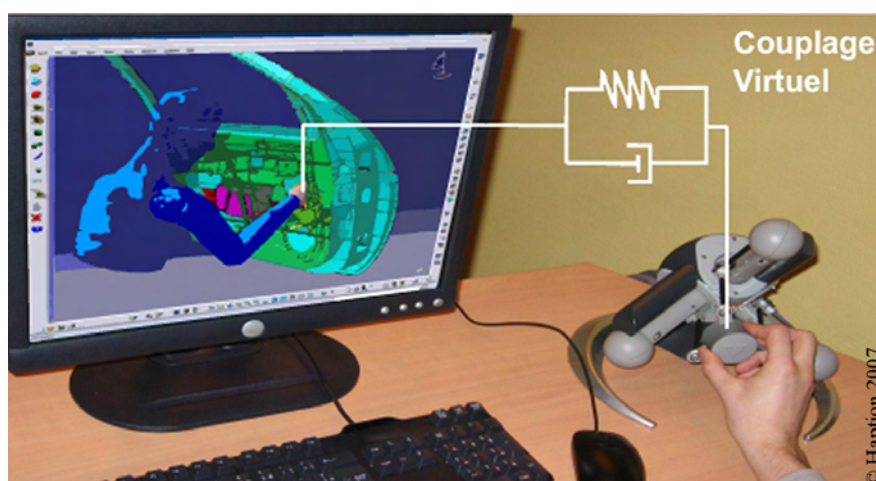


FIG. 1.4 – Introduction d'un couplage virtuel entre le dispositif haptique et l'objet virtuel du simulateur, lors d'une tâche d'assemblage CAO (courtoisie de Haption).

La théorie de la commande propose en effet des conditions nécessaires pour assurer la stabilité d'un système asservi, sous réserve que l'utilisateur comme l'environnement virtuel soient passifs, c'est-à-dire qu'ils n'ajoutent pas d'énergie mécanique au système. En se basant sur cette hypothèse, les auteurs proposent d'utiliser le couplage virtuel afin de garantir la stabilité globale du système, la simulation devant être contrôlée pour ne pas ajouter d'énergie au système, comme le prévoit la théorie de la passivité [MH03]. Mahvash *et al.* [MH05] étudient précisément les conditions nécessaires de passivité, notamment sur les aspects fréquentiels, afin de garantir la passivité du système. Ils proposent une méthode d'activation passive d'efforts locaux d'interaction, qui interdit les dépassements d'énergie potentielle dans l'environnement virtuel, généralement responsables des instabilités. Ils appliquent leurs modèles à des simulations haptiques faisant intervenir des objets déformables.

Adamas *et al.* [AH99] décrivent le concept de couplage virtuel pouvant s'appliquer tant à des modèles impédants qu'admittants en découplant la partie interface de la partie simulation. Mc Neely *et al.* [MPT99] proposent l'optimisation du paramétrage des raideurs virtuelles, par l'application d'heuristiques basées sur le nombre de contacts simultanés. De même, Zilles et Salisbury [ZS95] proposent une approche heuristique, nommée god-object, qui reconsidère l'état d'interpénétration au moment du contact afin de simplifier les lois de contrôle de l'interface. Ruspini *et al.* [RKK97] proposent l'utilisation d'un proxy virtuel, extension de ces god-objects, pour coupler une interface haptique avec un simulateur basé sur des contraintes portant sur trois degrés de liberté (3 ddl). Ortega *et al.* [ORC07] proposent une version à six degrés de liberté (6 ddl) de cette méthode god-objects

dans une application haptique pour des solides rigides. Otaduy *et al.* [OL06] utilisent le couplage virtuel, qui, couplé avec des algorithmes d'intégration implicite passifs, permettent d'obtenir des simulations haptiques stables à hautes fréquences.

Minsky *et al.* [MYS⁺90] explorent ces problèmes de stabilité, pointant les compromis à réaliser entre la fréquence de simulation, la raideur, la viscosité de l'interface ainsi que l'influence du bras de l'utilisateur qu'ils considèrent comme un système raide-visqueux. Afin d'évaluer ce dernier critère, les auteurs reprennent les travaux de [Hog88] sur l'impédance d'un bras, qui peut varier selon ses mouvements radiaux ou tangentiels. Lertpolpairoj *et al.* [LML02] découplent les problèmes liés au contrôleur de ceux de l'environnement, en proposant un couplage virtuel adaptable aux différentes fréquences d'interaction. Pour cela, ils intègrent l'impédance perçue par l'utilisateur dans leur paramétrage tout en assurant la stabilité du système. Renz *et al.* [RPP⁺01] étudient les problèmes de couplage pour différentes interfaces haptiques dans l'optique d'en améliorer la stabilité globale. Ils considèrent le couplage virtuel comme un filtre dynamique (*shaping filter*) qui réduit les forces de contact de telle manière que ni l'interface ni la boucle de force, qui inclut l'utilisateur, ne puissent devenir instables.

1.4 Dissociation d'une tâche d'assemblage interactive

Dans le cadre des simulations haptiques d'assemblage, nous proposons de dissocier une tâche élémentaire d'assemblage en deux étapes :

- une phase d'exploration ;
- une phase d'assemblage, que nous qualifierons de "fonctionnelle".

Dans une étape que nous nommons *exploratoire*, l'utilisateur ne s'intéresse qu'à la mise en interaction de l'objet qu'il pilote avec le reste de l'environnement fixe. Cette phase d'exploration consiste à laisser l'utilisateur découvrir l'environnement virtuel. Il y est libre de tous mouvements et peut interagir haptiquement avec les différents objets de la scène. Cette phase, qui est associée à une zone géométrique de l'espace, permet à l'utilisateur de se créer une représentation mentale (cognitive) visuelle et haptique de la scène. Dans cet espace, aucune tâche d'assemblage n'est prévue entre l'objet piloté et le reste de l'environnement.

La phase d'assemblage, que nous appellerons souvent *phase fonctionnelle* est en général celle qui justifie la réalisation de la simulation. Elle intervient après la phase exploratoire, en proposant la réalisation d'une tâche d'assemblage entre deux objets. En effet, quand deux objets sont dans des configurations spatiales qui rendent leur assemblage possible et/ou prévu, on considère la fin de la phase

exploratoire. Pour définir ces configurations, on peut basiquement considérer des "distances à la cible" entre l'objet piloté et l'endroit où il doit être assemblé. Durant cette phase, l'utilisateur utilise sa connaissance de l'environnement pour réaliser la tâche. C'est dans cette phase que l'on souhaite assister l'utilisateur dans la réalisation d'assemblage.

1.5 Conclusion sur l'état de l'art de l'haptique généralisée

Dans cette section, nous avons tout d'abord rappelé les notions liées au traitement des interactions haptiques au sein d'une simulation, en nous attachant à décrire sommairement des critères techniques nécessaires à l'amélioration d'une interface à retour d'effort. Nous avons alors détaillé le schéma de fonctionnement classique d'une simulation haptique, depuis le dispositif physique jusqu'au moteur de gestion des contacts chargé de retourner les efforts à l'utilisateur. Nous avons alors mis en évidence les différentes problématiques liées au couplage entre ces deux modules : une interface physique asservie soumise aux lois de l'automatique couplée à un simulateur physique chargé de restituer fidèlement des lois de comportement physique. Cette présentation a permis de souligner les problèmes de stabilité qui peuvent survenir au niveau de la simulation haptique et certaines solutions mises en place, industriellement et scientifiquement, pour y remédier.

La manipulation haptique d'objets CAO, dans le cadre précis des applications de réalité virtuelle, nous confronte ainsi au problème du choix de l'architecture de l'interface haptique, avec laquelle l'utilisateur doit réaliser des assemblages CAO. Les contraintes fortes liées à l'haptique, dans ce domaine, résident principalement dans la qualité du rendu haptique lors des interactions entre objets simulés. La stabilité du système, associée à la transparence de l'interface et la raideur des contacts ressentis, imposent à la simulation d'atteindre des performances de calcul en temps-interactif. Au sein d'une simulation interactive, le couplage entre la boucle haptique et la boucle de simulation physique est largement étudié dans la littérature afin de respecter au mieux, par des compromis, ces critères de rendu haptique.

Pour les simulations haptiques d'assemblages CAO, l'identification de ces contraintes nous permet de poser un certain nombre de choix d'architecture et de type de couplage, afin d'obtenir des rendus haptiques satisfaisants pour les tâches d'assemblage. En recentrant ces problématiques du rendu haptique, nous proposons de dissocier les tâches d'assemblage en deux parties. La première se concentre sur la manipulation haptique "classique" d'objet 3D, où l'exploration haptique des environnements est primordiale. La deuxième se concentre sur les tâches d'assemblage d'objets complexes, où l'utilisateur doit réaliser des opéra-

tions précises et prévisibles sur les objets CAO qu'il manipule.

Chapitre 2

Dynamique non régulière appliquée à l'haptique

2.1 Introduction

Dans ce chapitre, nous proposons un état de l'art sur le traitement des contacts et des chocs au sein de simulations dynamiques, en portant un regard particulier sur les approches pénalisées et non régulières de la dynamique et sur leur utilisation dans des applications haptiques interactives. Nous nous attachons dans un premier temps à décrire le processus d'interaction haptique, en décrivant les principes de mise en œuvre du retour d'effort ainsi que les solutions existantes gérant les simulations haptiques. Après une présentation des méthodes de détection de collision, nous présentons les modèles de traitement dynamique des efforts d'interaction entre objets virtuels, selon les approches par pénalisation et par dynamique non régulière. Nous soulignons pour cela les dépendances qui existent entre les méthodes de détection des collisions et les méthodes de détermination des efforts d'interaction. Nous proposons alors notre contribution sur l'utilisation des paradigmes de dynamique non régulière pour le calcul des efforts d'interaction inter-objets dans le cadre des simulations haptiques. À partir de l'approche *time stepping* de la dynamique non régulière, qui propose une modélisation rigoureuse des comportements mécaniques d'objets simples, nous présentons l'adaptation en temps-interactif des lois dynamiques existantes pour la simulation haptique. Le couplage haptique que nous avons mis en place est présenté à travers Conf&tis, une plate-forme expérimentale que nous avons réalisé dans le cadre de cette thèse.

2.2 Etat de l'art du traitement des interactions dans les simulations haptiques

Le logiciel de simulation physique est chargé de mettre à jour l'environnement ainsi que les objets virtuels simulés au cours du temps. Il prend en charge les modèles physiques des objets et calcule les relations qui apparaissent entre eux ainsi qu'avec leur environnement. Ces relations sont reliées aux informations calculées par le module de détection de collision. Les positions et vitesses des objets sont calculées, par le module de détermination des efforts d'interaction, par l'intégration temporelle des lois de la dynamique, qui nécessitent de connaître préalablement l'ensemble des efforts d'interaction qui existent entre eux (voir figure 2.1).

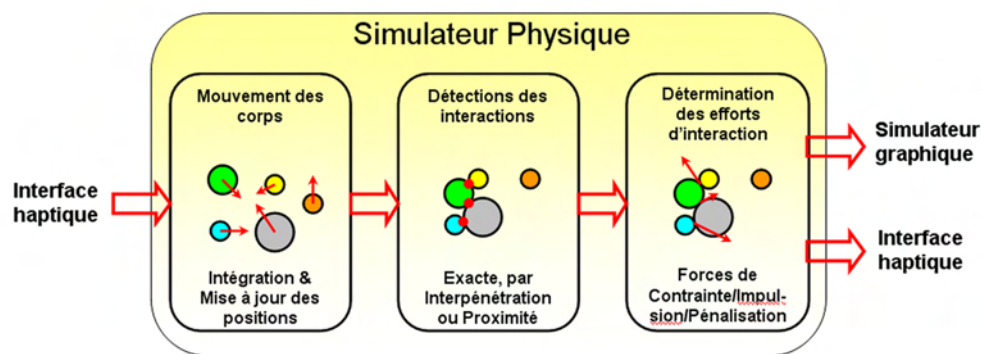


FIG. 2.1 – Schéma classique d'un simulateur physique.

2.2.1 Détection des collisions

L'objectif de la détection de collision est de repérer et de signaler un contact géométrique entre deux objets ou plus. Selon les cas, cette détection a lieu quand va se produire ou quand vient de se produire une collision. Les algorithmes de détermination des efforts se basent sur ces informations fournies par les détecteurs de collision pour mettre à jour les positions des objets simulés. Ces informations représentent en effet les configurations spatiales et/ou temporelles des contacts ainsi détectés, nécessaires aux calculs des efforts d'interaction.

On soulignera, en premier lieu, la diversité des paradigmes de représentation existants, chacun fournissant des informations de collision différentes : estimation du moment de la collision, distance et vitesse d'interpénétration ou champs de distances minimums entre objets (voir figure 2.2), et ce pour les différents types de géométrie à considérer : convexes ou concaves, polyèdres ou surfaces paramétriques etc. Nous invitons le lecteur à se référer aux travaux de [LG98], [JTT01]

et [KHI⁺07] qui dressent des bilans plus complets sur les différentes approches de la détection de collision, ainsi qu'aux travaux de [MKF03] qui soulignent les dépendances qui existent entre les paradigmes de détection et ceux de calcul des efforts.

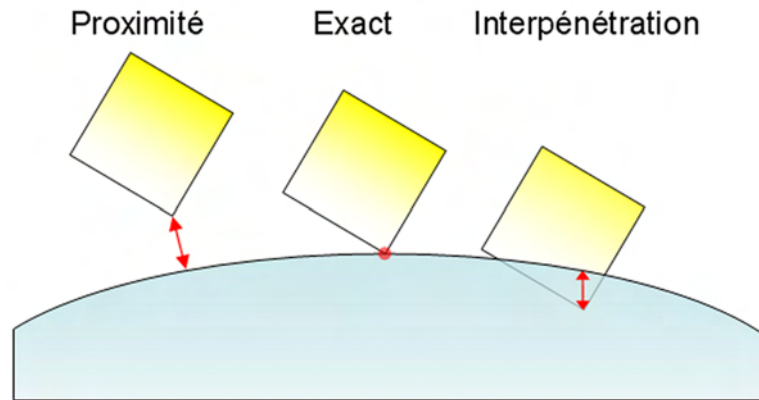


FIG. 2.2 – Différentes approches du contact entre solides rigides : par proximité, exacte ou par interpénétration.

D'un point de vue tribologique, un contact peut être considéré comme une déformation locale, élastique ou plastique, entre deux solides qui se touchent. D'un point de vue technique, la détection de collision s'effectue entre modèles géométriques numériques, discrétisés spatialement. Dans l'hypothèse d'objets non soumis à la déformation lors des contacts (solides rigides parfaits), la détection de collision revient à imposer une condition de contact, c'est-à-dire un état virtuel satisfaisant au mieux un besoin de simulation. Indépendamment de la nature des géométries en jeu, on peut ainsi considérer l'état du contact selon trois approches :

- la première consiste à considérer l'interpénétration entre les objets. Étant donné la nature discrétisée de l'environnement de simulation, on peut s'assurer de la présence d'un contact une fois que les objets ont pénétré les uns dans les autres. On peut dès lors considérer un contact comme une (faible) interpénétration entre modèles géométriques ;
- la deuxième consiste à considérer la proximité entre les objets. Plutôt que de considérer un contact que lorsque celui-ci a déjà eu lieu, il est possible de considérer la condition de contact comme une proximité. Deux solides distants d'une certaine distance seuil prédéterminée, sont suffisamment proches pour être considérés en contact (distance euclidienne entre polyèdres [LC91]) ;
- la troisième consiste à considérer l'instant et/ou l'endroit de la collision entre les objets. On s'attache à déterminer l'instant théorique exact de la collision. Le couple point/instant de contact est alors obtenu par calcul

plutôt que par des tests géométriques.

Afin de déterminer la configuration de contact, il existe deux types d'approches, dites discrètes ou continues. Les modèles de détection discrets permettent d'évaluer l'instant de la première collision, après que celle-ci ait eu lieu, puis, en revenant à l'instant précédant cette collision (*backtracking*), ils affinent le pas de temps en mesurant la profondeur d'interpénétration. Cette opération est alors répétée itérativement jusqu'à ce que l'intervalle de temps, dans lequel on considère que le contact a eu lieu, soit suffisamment précis. Une fois cet instant retrouvé, les objets sont (re)positionnés et le calcul des forces de contact peut alors être effectué. Ces méthodes, auxquelles nous pouvons rapporter des variantes par dichotomies (test de collision à réponse booléenne) [vGS01] ou par anticipation (identification de l'intervalle de temps durant lequel aucune collision n'est possible) [CK86], sont relativement courantes en simulation haptique bien qu'elles s'attardent à chercher dans le passé la configuration qui n'a pas violé la contrainte de pénétration.

Les modèles de détection continus, dits CCD (*continuous collision detection*), prédisent l'instant exact de la collision, empêchant ainsi l'interpénétration entre les objets, cette condition pouvant s'avérer nécessaire dans certaines applications telles que le *motion planning* ou la simulation chirurgicale. [RKC00] décrivent une méthode pour calculer algébriquement l'instant de la collision de deux objets polyédriques en mouvements relatifs. Pour ce faire, ils calculent des prévisions de mouvements en les interpolant entre deux pas de temps, résolvant simplement une équation polynomiale de troisième ordre. [ZRLK07] se sont intéressés aux modèles articulés, reprenant les techniques de *arbitrary in-between motion* pour interpoler les mouvements de solides liés par des liaisons, entre deux pas de temps successifs, puis vérifier d'éventuelles collisions sur ces trajectoires.

L'obtention des informations de collision doit répondre à des critères de performance temporelle et de précision, liés à la simulation interactive. Au sein d'une simulation, la part de calcul allouée à la détection de collision est d'autant plus importante que la scène virtuelle est complexe. La rapidité de cette étape dépend, par exemple, du nombre de tests géométriques de bas niveau à effectuer entre les primitives géométriques des objets, nombre qui est d'autant plus important qu'il y a d'objets. Les tests d'intersection de polygones, pour ne citer qu'eux, nous mettent face à une problématique liée aux temps de calcul. Pour des modèles numériques complexes pouvant compter des milliers voire des millions de polygones, il n'est plus optimal de réaliser systématiquement tous les tests entre leurs éléments (nuages de points, surfaces paramétriques, triangles...).

Il existe ainsi un vaste champ de recherche dans le domaine de la détection de collision, dont les résultats ont permis de créer des algorithmes utilisés industriellement : I-COLLIDE [CLMP95], V-COLLIDE [HLC⁺97], SOLID [vdB99]

et CONTACT [RKC01]. Selon le degré de précision attendu, il est possible de restreindre l'espace combinatoire de ces tests par l'utilisation de méthodes d'accélération d'accès aux objets potentiellement en collision :

- par tests de collision entre volumes englobants [GLM96] ou par découpage en domaines spatiaux (domaines de Voronï [EL00]) ;
- par approximation de l'information de collision selon des approches évolutionnistes [Jou06] ;
- par des méthodes hybrides pour les objets déformables [TKH⁺04], par hiérarchies de volumes englobants (de type BD-Tree [JP04] ou AABB-Tree [vdB98]), par approches stochastiques [KNF04] ou par évaluation de distance d'interpénétration [FL01] ainsi que des méthodes spécifiques aux polyèdres convexes (méthode GJK dans [GJK88]).

2.2.2 Détermination des efforts d'interaction

Le contact entre objets peut être décrit par des modèles physiques, c'est-à-dire par l'utilisation de lois de la physique pour simuler de manière réaliste les comportements des objets [BB88]. Le traitement des interactions se traduit alors par la réponse, en termes de dynamique, aux informations de contact délivrées par la détection. Dans cette section, nous expliquons plus précisément la manière dont on exploite ces informations pour mettre à jour les positions des objets et les éventuels retours d'effort. Nous regardons de plus près les bases mécaniques de la détermination des efforts d'interaction, puis nous étudions deux types de modèles dynamiques : les modèles dits pénalisés et les modèles non réguliers.

À la base des simulateurs à modèles physiques, les positions et les vitesses des objets simulés, que l'on considère comme rigides ou comme déformables, sont calculées par intégration des équations différentielles de la dynamique. Ces équations, issues du principe fondamental de la dynamique (PFD) expriment la relation qui existe entre la masse d'un corps, son accélération et les efforts qui lui sont appliqués. Cette approche par la mécanique du solide permet ainsi de décrire l'évolution des solides en fonction du temps. Pour un solide rigide de masse m , de matrice d'inertie $[\mathbf{I}]_{\mathbf{O}}$ exprimée en un point $\mathbf{O}$ quelconque, d'accélération $\vec{\mathbf{a}}$ et d'accélération angulaire $\vec{\dot{\omega}}$ à l'instant t , les équations de la dynamique s'écrivent :

$$\begin{cases} m \vec{\mathbf{a}} = \sum \vec{\mathbf{F}}_{ext} \\ [\mathbf{I}]_{\mathbf{O}} \vec{\dot{\omega}} = \sum \vec{\mathbf{C}}_{ext} \end{cases} \quad (2.1)$$

Avec $\sum \vec{\mathbf{F}}_{ext}$ la somme des forces extérieures appliquées au système, et $\sum \vec{\mathbf{C}}_{ext}$ la somme des couples extérieurs exprimés au même point $\mathbf{O}$ du système. Dans les simulations où interviennent des contacts entre solides, la résultante $\vec{\mathbf{R}}$ des

efforts de contact s'additionne et conduit, pour la partie translation de l'équation [2.1], à l'équation [2.2] :

$$\mathbf{m} \vec{\mathbf{a}} = \sum \vec{\mathbf{F}}_{ext} + \vec{R} \quad (2.2)$$

Dans les cas de contact (ou choc) entre solides, la détermination de ces efforts constitue le point dur du problème du traitement haptique. La détermination des efforts locaux d'interaction peut, dès lors, être réalisée par l'utilisation de lois régularisantes telles que dans les approches par pénalisation, ainsi que par des lois non régulières.

2.2.3 Modèles pénalisés

La méthode dite de pénalisation est une méthode simple qui permet de calculer les efforts qui s'exercent sur des objets en contact. Pour un contact donné, elle représente la force répulsive qui s'exerce entre les objets en contact. Cette force est issue de l'introduction entre les objets, d'un ressort virtuel. L'expression de cette force est alors calculée en appliquant l'expression suivante :

$$\vec{\mathbf{F}} = -\mathbf{k} \cdot (\mathbf{l} - \mathbf{l}_0) \cdot \vec{\mathbf{n}} \quad (2.3)$$

Dans cette expression, $\vec{\mathbf{F}}$ est la force normale de répulsion engendrée par la déformation du ressort, $\mathbf{k}$ est la constante de raideur du ressort virtuel, $\mathbf{l}$ est la longueur courante du ressort égale soit à la profondeur d'interpénétration, soit à la distance critique entre les deux objets, $\vec{\mathbf{n}}$ est le vecteur normé qui s'appuie sur le couple de points en interaction, $\mathbf{l}_0$ est la longueur à vide de ce ressort qui peut soit être égale à zéro si on considère une interpénétration, soit être égale à une distance seuil (non nulle) si on considère une détection par proximité [JW03]. La résultante normale des efforts de contact est ici considérée comme explicite en fonction des positions, du fait qu'on ne considère pas la discontinuité de la vitesse au moment du contact. La composante tangentielle de cette force est généralement introduite par l'utilisation de lois de frottement, que nous explicitons dans la section 2.2.5. Dans cette équation, on peut également, si le comportement des objets le nécessite, ajouter des termes de viscosité de la forme $-\mathbf{C} \cdot \dot{\mathbf{l}} \cdot \vec{\mathbf{n}}$, où $\mathbf{C}$ est le coefficient de viscosité.

Les méthodes par pénalisation sont largement exploitées dans les simulations dynamiques pour calculer des efforts de contact. Moore *et al.* [MW88] furent parmi les premiers à les exploiter dans une simulation dynamique. D'autres études furent réalisées depuis ; on retiendra l'utilisation de raideurs exponentielles par McKenna [MZ90], une comparaison avec les modèles analytiques par Keller [KSZB95], ainsi qu'une étude de Hasegawa [HS04] qui dresse un bilan de ces types d'approches. Luo et Xiao dans [LX04] reconsidèrent les configurations géométriques de contact et les vitesses relatives des objets afin d'ajouter des

composantes inertielles aux efforts d'interaction. Pour cela la méthode propose la détection temps-réel des interactions entre les surfaces des solides rigides préalablement traitées (représentation implicite et/ou paramétrique). Puis après avoir proposé un typage des contacts qui intègre les configurations en vitesse des objets en mouvement, ils appliquent un modèle pénalisé avec frottement, afin de calculer les efforts d'interaction.

Ces méthodes de traitement par pénalisation sont parmi les plus utilisées dans les simulations haptiques car elles sont facilement implémentables. Le traitement des efforts de contact offre des performances temporelles pouvant être considérées comme convenables pour le rendu haptique. Toutefois, la majorité de ces méthodes requiert l'interpénétration des modèles pour le calcul des efforts d'interaction ou l'approximation du phénomène de contact. Ceci peut s'avérer problématique dans des applications telles que la chirurgie [LCN99]. Si ces méthodes sont simples et pratiques à mettre en œuvre pour des applications en temps-interactif avec retour d'effort, elles ne constituent qu'un modèle physique approché. [HFKS03] soulignent que le réglage de la constante $\mathbf{k}$ de raideur du ressort virtuel ne peut être obtenu par des paramètres mécaniques, et est généralement ajusté en fonction des conditions de contact. Alors que la fréquence de calcul des efforts est limitée par la fréquence d'échantillonnage de la simulation, comme le prévoit le théorème de Nyquist-Shannon, la raideur de l'équation différentielle est fonction croissante de la constante de raideur du ressort. Certains solveurs numériques explicites, chargés de résoudre les équations différentielles de la dynamique, posent des problèmes de stabilité liés à la taille du pas de temps quand cette constante $\mathbf{k}$ devient trop importante.

2.2.4 Exemples industriels basés sur le modèle pénalisé

En dépit des approximations mécaniques réalisées par le modèle pénalisé, les avantages proposés en termes de temps de traitement et rendu haptique sont considérés comme acceptables pour beaucoup d'applications industrielles. Afin d'illustrer l'approche par pénalisation, nous présentons deux exemples concrets de traitement des interactions utilisés dans des moteurs haptiques industriels. Ces deux modèles, baptisés LMD et VPS, appliquent une raideur virtuelle à l'endroit du contact afin d'en déterminer l'effort d'interaction. Ils utilisent chacun des moteurs de détection originaux, qui retournent des informations de contact différentes et illustrent la dépendance qui existe entre les méthodes de détection et de détermination des efforts d'interaction.

Le modèle LMD (*local minimum distance*) Johnson *et al.* [JW03] décrivent une méthode complète de détection et de traitement des collisions basée sur les distances locales minimales entre les objets géométriques. Pour les modèles po-

lygonaux, cette méthode évalue la distance entre deux objets et fixe un seuil de collision. Cette approche par proximités, développée dans [JWC05], se base à la fois (voir figure 2.3) :

- sur des techniques de recherche de distance euclidienne entre modèles. Cette distance est calculée entre des nœuds spécifiques à chaque hiérarchie de volumes englobants les objets [LGLM99] ;
- sur la recherche, en parallèle, de distances locales entre géométries élémentaires, par descente de gradient des LMD [JW04] ;
- sur la recherche de distances minimums globales entre deux surfaces paramétriques pour les polytopes complexes [PML95].

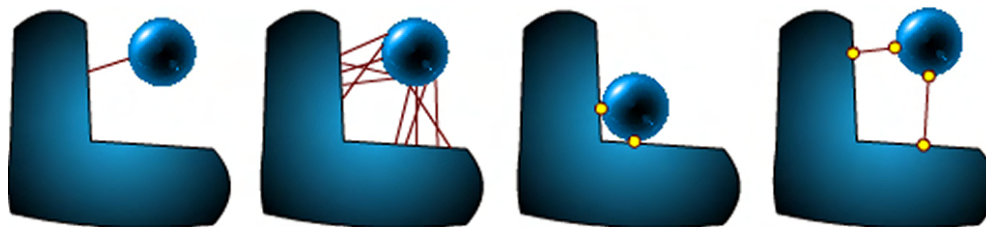


FIG. 2.3 – De gauche à droite : distance minimale globale - paires en-dessous d’une distance seuil - points de contact - distances minimales locales (d’après (Johnson *et al.*, 2003)).

Afin de pallier les problèmes de combinatoire, les auteurs décomposent les géométries en hiérarchies de sphères englobantes [JC01], puis effectuent des tests de proximité par des mesures vectorielles entre les différents nœuds. La proximité critique est définie comme une distance minimum en-dessous de laquelle les objets seront considérés comme en contact. Dès lors, plutôt que de déterminer les efforts de contact une fois que les objets sont entrés en collision, le modèle applique des forces préventives de répulsion dès que les modèles sont suffisamment proches. Pour cela le système applique un modèle pénalisé, en considérant chaque distance minimum locale de longueur critique comme un ressort virtuel dont les attaches sont représentées par la paire des nœuds considérés. Du point de vue du rendu haptique, la gestion des interactions ainsi traitées donne satisfaction. De plus, l’implémentation est aisée aussi bien pour la gestion par pénalisation que pour le moteur de détection. L’utilisation de champs de distances cohérents d’un pas de temps à l’autre permet également de pallier les problèmes de stabilité. Ce type de solution est appliqué dans l’industrie et dans la recherche, par exemple dans le domaine de l’énergie nucléaire au CEA (commissariat à l’énergie atomique).

Le modèle VPS (*voxmap-pointshell*) [MPT99] proposent un autre modèle de traitement des interactions spécifiquement destiné à des applications hap-

tiques. Dans cette approche, la détection et le traitement des interactions ne sont pas directement effectués sur les surfaces des objets, mais sur des discrétisations de leurs géométries :

- la surface de l'objet candidat est représentée par un échantillon de points nommé PointShell. À chaque pas de temps, ces points sont mis à jour en fonction des transformations de l'objet puis associés aux normales locales de la surface (voir figure 2.4). Une technique de ce type d'échantillonnage par voxelisation est décrite dans [MPT06] ;
- l'objet antagoniste est décomposé par une méthode de voxelisation, en cubes élémentaires organisés dans une carte indexée appelée Voxmap.

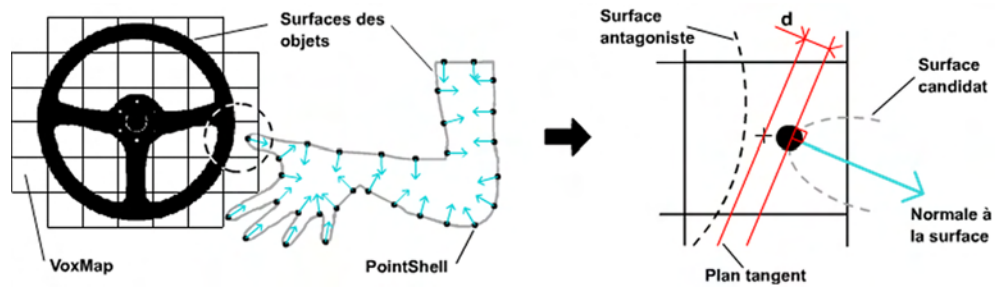


FIG. 2.4 – Distance de proximité d , entre le PointShell de l'objet candidat et la Voxmap de l'objet antagoniste en contact (d'après (McNeely *et al.*, 2006)).

Pour des objets rigides [RPP⁺01] ou pour des objets déformables [BJ07], la détection des collisions revient alors à tester l'appartenance des points aux voxels. L'information de contact est calculée comme une distance d'interpénétration entre ces éléments. Les cas possibles de détection sans interpénétration sont traités comme des distances nulles. La force de contact locale est alors calculée en utilisant une méthode de pénalisation (équation [2.3]), c'est-à-dire proportionnellement à cette distance d'interpénétration. L'effort global d'interaction entre les deux solides est finalement calculé comme étant la résultante du champ des forces locales précédent. Ce type de solution, développé par BoeingTM, est appliqué industriellement, par exemple dans l'unité de commande IPSI-Serveur des bras Haption VirtuoseTM.

2.2.5 Lois de contact

Afin d'assurer un comportement simulé qui représente au mieux la réalité, les différentes approches non régulières appliquent des lois de comportement physiques. C'est ainsi que la condition de non-pénétration de Signorini, la loi de rebond dite de Newton et la loi de frottement de Coulomb viennent ajouter des conditions supplémentaires au système.

Le fait que deux solides ne peuvent s'interpénétrer peut être exprimé par la condition de non-pénétration de Signorini. Celle-ci traduit le caractère unilatéral des liaisons de contact (ou de choc). En cas de contact, ces deux solides ne doivent pas pouvoir traverser leurs surfaces respectives ni s'attirer mutuellement (voir figure 2.5). Cette condition s'écrit :

$$g \geq 0 \quad r_n \geq 0 \quad g \cdot r_n = 0 \quad (2.4)$$

Dans cette équation, g est l'écartement relatif entre les solides et r_n est la composante normale de l'effort de contact. Cette condition traduit donc que si les solides sont éloignés ($g \geq 0$), alors la force de répulsion qu'exerce un solide sur l'autre est nulle ($r_n = 0$). S'ils sont en contact ($g = 0$), alors la force de répulsion est positive ou nulle ($r_n \geq 0$).

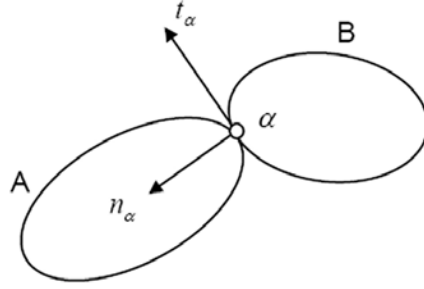


FIG. 2.5 – Contact entre deux solides A et B, en un point α , dont l'effort de contact est décomposable en une composante normale n et une composante tangentielle t .

[DAK04] proposent de formuler les déformations des modèles par l'application du problème de Signorini. Ils utilisent un moteur de détection discret pour déterminer, en premier lieu, des configurations de premier contact entre solides rigides et déformables. Dans l'espace du contact, les efforts de contact sont alors déterminés par une formulation en LCP du problème de Signorini. Par des inégalités et des relations complémentaires, ils proposent de formuler la distribution des efforts de chaque zone de contact, ainsi que la relation entre ces efforts et les déplacements des objets déformables (selon le type d'interpolation entre l'espace des contacts et les déplacements des éléments finis). En parallélisant le processus de calcul de la déformation, ils proposent une application haptique gérant le contact entre un objet déformable et un objet rigide, et dans [DDKA06] entre deux objets déformables. Dans cette dernière étude les auteurs précisent la gestion des contacts frottants, en utilisant une approximation pyramidale du cône de frottement de Coulomb telle que décrite dans [ST96], et en résolvant le LCP par une adaptation de l'algorithme Gauss-Seidel.

Le frottement peut être traité en utilisant le modèle de Coulomb, classique, qui stipule que la norme de la composante tangentielle de l'impulsion locale doit rester inférieure à la valeur de la norme de l'impulsion normale multipliée par le coefficient de frottement μ :

$$\|R_t\| \leq \mu R_n \quad (2.5)$$

Avec R_t et R_n , les composantes, tangentielle et normale, de la force de frottement. Si on considère le glissement, cette force se trouve à la frontière du cône de Coulomb ($\|R_t\| = \mu R_n$). Afin de traduire cette relation en un problème de complémentarité, le cône de Coulomb est linéarisé au point de contact par une approximation pyramidale [SS98]. La réalité de ce modèle peut toutefois être contestée en plusieurs situations telles que dans les cas de roulements frottants. Il est possible de traiter le cône de frottement de Coulomb sans en proposer d'approximation pyramidale pour corriger les effets de dérive lors de cyclages en effort sur des corps soumis à du frottement [RAD05].

La formulation de Signorini (équation [2.4]) peut être formulée en termes de vitesses, en notant v_n la vitesse normale [Jea99] :

$$\begin{aligned} &\text{Pour } t_0 \text{ tel que } g(t_0) \geq 0, \\ &\forall t \in I, \text{ si } g(t) \leq 0, \text{ alors } v_n(t) \geq 0, \quad r_n(t) \geq 0, \quad v_n(t).r_n(t) = 0 \end{aligned} \quad (2.6)$$

Elle complète alors la loi de choc de Newton généralisée, formulée par Moreau, qui sert à modéliser le rebond, en particulier pour les cas de multicontacts. Un rebond s'exprime ici par le fait que la vitesse normale avant le choc est égale à la vitesse normale après le choc, au coefficient de restitution e près. Sa valeur est constante et comprise dans la plage 0 (choc mou) et 1 (choc élastique). Ce modèle représente la caractéristique locale de l'interaction, qui dépend des matériaux en présence et de la géométrie locale. Excepté dans des configurations de type sphère sur sphère, elle est établie de manière empirique :

$$v_n^+ = -e_n v_n^- \quad (2.7)$$

Cette équation est valable quand on peut définir un plan d'interaction, par exemple pour des corps sphériques. Pour des corps à géométrie plus complexe, le coefficient de restitution e n'a plus vraiment de sens et il faut prendre en compte la géométrie telle que l'étudient Lessaux *et al.* [SCM04]. Un exemple nous amène à nous interroger sur la validité de cette loi. Considérons la configuration particulière d'un bloc parallélépipédique oscillant sur un sol plan. L'équation [2.7] n'autorise pas le décollement au passage de l'objet par le statut de double contact. La transformation de l'équation [2.7] en une inégalité appliquée à tous les points

de contact autorise, dans ce cas, le décollement qui est plus conforme à la réalité [RAD05] :

$$v_n^+ \geq -e_n v_n^- \quad (2.8)$$

2.2.6 Modèles non réguliers (*non-smooth dynamics*)

Les modèles pénalisés proposent de déterminer les efforts d'interaction en exprimant explicitement (équation [2.3]) les efforts à partir des paramètres de configuration. Les modèles non réguliers sont basés sur le traitement de problèmes complémentaires comme indiqué par [Löt82] ou par [Mor88] qui propose la méthode CD (*contact dynamics*). La condition de non pénétration entre les objets, modélisés par des solides rigides, est formalisée par des inéquations qui expriment le caractère unilatéral de l'interaction. Les multiplicateurs de Lagrange associés représentent alors les forces de contact. Ces méthodes non régulières, dites *non smooth*, diffèrent donc des méthodes pénalisées par le traitement des forces de contact qui ne sont plus explicitement exprimées en fonction des paramètres de configuration mais traitées comme des multiplicateurs associés aux contraintes unilatérales. La méthode NSCD (*non smooth contact dynamics*) [Jea99] propose des fondements algorithmiques qui permettent de simuler des ensembles de solides soumis à de multiples contacts. Dans cette méthode, les différentes lois de comportement telles que les lois de frottement, les lois d'adhésion ou les lois d'impact sont traitées numériquement sans régularisation. Cette méthode utilise des algorithmes itératifs, tel que les algorithmes de type Gauss-Seidel, pour résoudre les problèmes non linéaires locaux. Ces non-linéarités s'expriment en termes de spatialité puisque les conditions géométriques de non-pénétration conduisent à traiter des inégalités. Elles s'expriment également en termes de temporalité puisque les chocs entre les corps créent des discontinuités de vitesse. Un état de l'art global sur les algorithmes de simulation numérique, dans le cadre de ces modèles non réguliers, a été proposé depuis par [BDP⁺02].

2.2.7 Approche par événements (*event-driven*)

Introduite par [Löt81], cette méthode de pilotage par événements cherche à déterminer le temps du premier contact, celui-ci pouvant survenir entre deux pas de temps de calcul [Löt82]. L'instant de collision est déterminé en effectuant des techniques de retour arrière (*backtracking*) [MW88]. Pratiquement, le simulateur effectue des affinements itératifs du pas de temps afin de déterminer l'instant de contact. Selon le degré de précision souhaité, le pas de temps est progressivement affiné, pouvant ainsi conduire à des temps de calcul importants. Par ailleurs, la recherche de cet instant doit être effectuée pour chaque contact, impliquant

d'autres variations du pas de temps et avec autant de cycles itératifs qu'il y a de contacts. Ces retours arrière étant coûteux dans le cas de collisions multiples, il est nécessaire de dissocier le traitement des contacts de celui des chocs. Cette discrimination se fait en comparant les vitesses relatives des solides à un certain seuil : en-dessous de ce seuil on considère un contact, au-dessus on considère un choc.

Cette approche est par ailleurs adaptée aux collisions multiples, en commençant par traiter les chocs qui ont le plus d'effet sur le mouvement des solides, puis en traitant les contacts. [KEP05] traitent la gestion des contacts multiples en utilisant leurs informations de masse, vitesse et localisation à l'instant de plus forte compression, afin de contraindre les solides rigides en vitesse.

[Bar89] propose une méthode analytique de calcul des forces de contact pour des systèmes de solides rigides en mouvement. Le simulateur proposé détecte les collisions potentielles (au sens de l'interpénétration) entre deux solides, puis le cas échéant calcule explicitement les forces à appliquer pour les séparer [Bar94]. Le système de contraintes ainsi appliqué garantit la non-interpénétration et satisfait les lois de la dynamique au sens de Newton. Analytiquement, ces forces de collision sont des impulsions.

D'autres approches parallèles à base d'impulsions [Mir00] détectent l'instant précis de la collision plutôt que de traiter le contact courant. Une fois cet instant de premier contact détecté, la résolution du système d'équations de la dynamique est permise par l'application d'une force artificielle, dite impulsienne, à l'endroit du contact. Diminuant le pas de temps en fonction du nombre de contacts, cette approche s'inscrit assez mal dans une simulation de type réalité virtuelle bien qu'elle gère les contacts et les chocs de la même manière [MC95]. Que l'on considère un contact ou un choc, l'effort de contact représente une inconnue supplémentaire qui interdit toute résolution explicite du système d'équations de la dynamique. On résout ce problème en appliquant au contact la loi d'impact de Newton, éventuellement associée à la loi de Coulomb dans le cas frottant. Pour résoudre le système, [Bar89] propose d'utiliser des heuristiques basées sur des techniques de programmation linéaire, puis considère le problème comme un LCP (*linear complementarity problem*), système d'inéquations traduisant les conditions de vitesses et de forces pour traiter la possibilité de décollement [Bar91]. [PPG04] utilisent une approche semblable, appliquée à des objets quasi rigides par l'utilisation de LCP élastiques. Les modèles déformables sont abordés par [JP01] et [PvdDJ⁺01] qui utilisent des CMA (*capacitance matrix algorithms*) et des fonctions de Green afin de déterminer des contraintes de déplacement sur les solides en contact.

En définitive, ces approches pilotées par les événements déterminent avec précision l'instant du choc, retournant ainsi des résultats acceptables pour les simulations interactives. Le problème réside dans le fait que ces approches deviennent très coûteuses lorsqu'il y a beaucoup d'interactions à traiter simultanément. En

effet, cette méthode nécessite un raffinement itératif du pas de temps pour rechercher l'instant de la collision ; et dans le cas où les interactions sont nombreuses, les temps de calcul sont peu appropriés à une simulation interactive [Dum05].

2.2.8 Approche temps-contrôlé (*time-stepping*)

L'approche temps-contrôlé (*time-stepping*) de la dynamique non régulière prend en compte les discontinuités de vitesse des contacts, en calculant la moyenne des efforts d'interaction sur un pas de temps, puis en la réintroduisant dans une formulation des équations de la dynamique. [ABD⁺99], [RDA04] et [Sau04] ont proposé cette approche pour des scènes constituées de multiples objets. Contrairement aux approches pilotées par les événements, le pas de temps est maintenu constant.

Cette approche consiste à utiliser des algorithmes de mécanique non régulière [Mor94], dans une approche continue en temps où le choc et le contact sont traités dans l'espace des vitesses de manière unifiée [LW05]. Elle repart de l'équation de la dynamique [2.1] et lui préfère une forme au sens des mesures exprimée en vitesse :

$$M(q)d\dot{q} = Q(q, \dot{q}, t)dt + \mathcal{L}d\nu. \quad (2.9)$$

Dans cette équation, dt est la mesure de Lebesgue, $d\dot{q}$ est la mesure d'accélération et $d\nu$ une mesure non négative pour laquelle $d\dot{q}$ possède une densité de mesure [Ren04] et $\mathcal{L}$ est la densité d'impulsion de contact. Cette approche introduit, sur un intervalle $]t^i, t^{i+1}]$, la quantité inconnue qui est une impulsion moyenne de contact :

$$\mathcal{L}(i+1) = \frac{1}{dt} \int_{]t^i, t^{i+1}]} \mathcal{L}d\nu.$$

C'est cette variable généralisée qui représente l'effort d'interaction et qui permet de déterminer la vitesse des objets $\dot{q}(i+1)$ après les chocs (celui-ci a lieu entre t^i et t^{i+1}).

Afin de déterminer les efforts de collision puis de mettre à jour les positions des objets, le problème est formulable en un LCP [Ani03], sous réserve d'approximations du cône de frottement [Glo99]. La résolution du problème revient alors à utiliser des méthodes numériques itératives [Mab98] ou par pivots [CPS92]. Renouf [Ren04] s'inspire de techniques propres aux milieux granulaires pour résoudre ce problème (MD, GEM et NSCD). Il revient d'ailleurs sur les différentes méthodes numériques existantes [RA06] : algorithmes itératifs (Gauss-Seidel, gradient conjugué projeté) ou direct (Lemke, programmation quadratique), et les compare avec la méthode NSCD en termes de complexité, convergence et temps de calcul. [Ste98] et [APS99] proposent une étude détaillée

de la convergence des méthodes de type time-stepping pour les solides rigides soumis à du frottement.

Cette méthode basée sur le contrôle du temps a l'avantage de ne pas discriminer les chocs des contacts. Le pas de temps, sous réserve qu'il soit suffisamment faible, peut être conservé constant tout au long de la simulation. Par contre, nous n'avons accès qu'à l'impulsion moyenne, qui sera d'autant mieux calculée que le pas de temps dt sera petit et nous n'avons pas accès à la configuration exacte d'interaction. De plus, s'il n'y a aucune interaction (cas libre) le schéma numérique est d'ordre 1, qui est moins précis qu'un schéma d'ordre plus élevé comme on peut en utiliser dans l'autre méthode (*event-driven*). Ces approches ont prouvé leur pertinence pour la simulation de systèmes granulaires à interactions multiples (ballast, mécanique des sols), mais n'ont jusqu'alors pas été introduites dans des simulations haptiques.

2.2.9 Conclusion sur l'état de l'art des modèles de traitement des interactions haptiques

Dans cette section, nous avons passé en revue différentes méthodes de traitement des interactions par l'utilisation de la dynamique non régulière. Dans les deux principales approches présentées, nous avons discuté des atouts et des faiblesses au regard de la simulation haptique selon des critères de vitesse de calcul, de nombre de contacts simultanés traités et de pertinence des résultats. L'approche par événements (*event-driven*) s'attache à simuler une force de réaction une fois que le contact a eu lieu. Puis en corrigeant les positions des objets *a posteriori*, elle modifie le pas de temps tout en ne gérant qu'un nombre limité de contacts simultanés. L'approche par temps-contrôlé (*time-stepping*) permet d'assurer une continuité du temps en considérant les efforts de contact comme des inconnues, auxquelles sont adjointes les lois qui régissent le comportement des contacts et des chocs.

2.3 Contribution : intégration de la NSCD dans une simulation haptique

Pour étudier l'utilisation des paradigmes de dynamique non régulière dans la simulation haptique, nous avons développé un simulateur dynamique dédié. Ce simulateur traite en particulier les contacts, les collisions et les frottements pour des applications de type RV. Ces travaux, sont présentés dans [TDAP09]. Le prototype fonctionnel ConF&TiS est la mise en œuvre de l'approche temps-contrôlé de la dynamique non régulière, qui permet la simulation d'environnements multiobjets en interaction entre eux. Développé à partir des travaux de [RAD05], ce

prototype a pour objectif la représentation des phénomènes réalistes dans la simulation d'objets virtuels, en se basant sur une application fidèle des phénomènes mécaniques. Ce simulateur permet de faire interagir un utilisateur avec des objets simulés et permet d'expérimenter la généricité entre l'interface et le simulateur.

Le simulateur traite en particulier des méthodes d'intégration temporelle pour des systèmes non réguliers et en gère la résolution numérique en prenant en compte les contraintes unilatérales, les frottements secs (loi de Coulomb) ainsi que les impacts formalisés par des problèmes de complémentarité. Ce prototype fonctionnel sert de plate-forme expérimentale à l'approche temps-contrôlé pour l'étude du rendu haptique en termes de vitesse, raideur, transparence et stabilité.

2.3.1 Conf&Tis : plate-forme de simulation haptique par dynamique non régulière

ConF&TiS, aussi noté Confetis, est un prototype fonctionnel développé par les équipes BUNRAKU (anciennement projet SIAMES)¹ et BIPOP² de l'IRISA, à partir des travaux de Renouf [RAD05]. Ce prototype a pour objectif la représentation des phénomènes réalistes dans la simulation d'objets virtuels, en se basant sur une application fidèle des phénomènes mécaniques. Il permet de faire interagir un utilisateur avec des objets simulés et permet d'expérimenter la généricité :

- entre l'interface et le simulateur ;
- entre l'algorithme de détection de collisions et la méthode de traitement des interactions.

Le simulateur traite en particulier des méthodes d'intégration temporelle pour des systèmes non-réguliers et en gère la résolution numérique en prenant en compte les contraintes unilatérales, les frottements secs (loi de Coulomb) ainsi que les impacts formalisés par des Complementarity Problems (LCP et MCP).

Le moteur de détection des collisions, baptisé N2, est issu de travaux (non publiés) de l'équipe Bunraku. Il s'emploie à déterminer des configurations de contact entre des primitives élémentaires (segment, plan, point) des objets. La méthode analytique qu'elle emploie est coûteuse, mais offre des performances acceptables dans la mesure où nous travaillons sur des volumes élémentaires (cubes, sphères, cylindres et plans).

La détermination des efforts d'interaction suit, quant à elle, une approche en temps-contrôlé de type NSCD, telle que nous l'avons décrite à la section 2.2.8.

L'intégration des méthodes de contact dans des applications de réalité virtuelle passe enfin par la possibilité de traiter des systèmes d'objets rigides polyarticulés.

¹IRISA-projet Bunraku - www.irisa.fr/bunraku, équipe projet commune Inria, CNRS, Université de Rennes 1, Insa de Rennes, École Normale Supérieure de Cachan.

²www.inrialpes.fr/bipop

Afin de répondre à des exigences de temps-réel, le nombre d'objets simulés doit être limité, toutefois des expériences [RAD05] ont montré la gestion correcte, du point de vue temporel, de 160 sphères.

Utilisant initialement la librairie graphique d'ODE ³, Drawstuff, Confetis à depuis été porté dans la plate-forme OpenMask (librairie graphique OpenSG), puis implémenté dans une version *stand-alone* utilisant la librairie OGRE 3D ⁴. C'est cette version, qui propose un moteur de visualisation et d'interface, qui nous sert de plate-forme expérimentale [TD08].

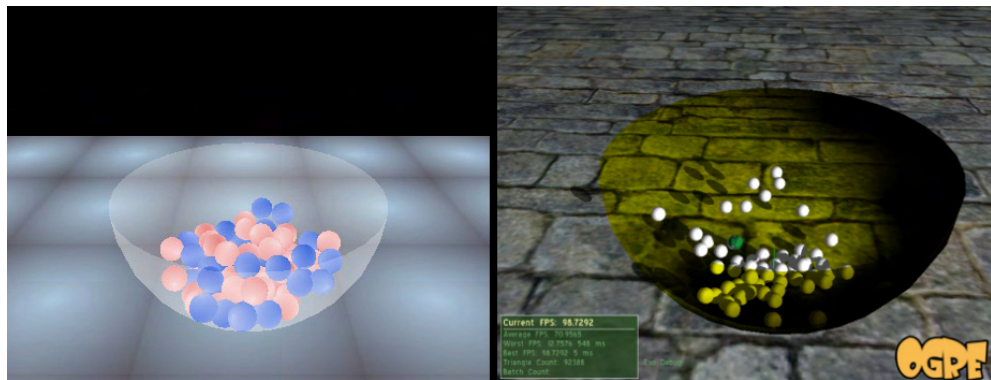


FIG. 2.6 – Visualisation des données sous OpenMask (à gauche) et Ogre (à droite).

Dans un premier temps, Conf&tis appelle ses fichiers d'initialisation qui contiennent en particulier :

- la description des objets : géométries, positions et vitesses initiales, densité du matériau ;
- les contraintes cinématiques imposées à ces objets : détermination de contraintes pour chaque degré de liberté (ddl) ;
- les paramètres des lois de frottement et de restitution ;
- les paramètres de la simulation et des solveurs numériques : pas de temps, nombre d'itérations, nombre de faces pour les approximations pyramidales (algorithme de LEMKE ou QP) ainsi que les paramètres propres à la méthode NLGS (Non Linear Gauss-Seidel), le cas échéant.

Ce simulateur dynamique fonctionne en admittance, c'est-à-dire que les données d'entrée sont des positions et les données de sortie sont des efforts. Toutefois la passivité, qui assure la stabilité du système, ne peut être garantie sans l'utilisation de couplages virtuels. Notre champ de recherche porte donc sur l'étude des compatibilités fréquentielles entre les différents blocs de notre modèle hap-

³Open Dynamic Engine - www.ode.org

⁴www.ogre3d.org/

tique, par le découplage de l'asservissement de l'interface physique du simulateur physique.

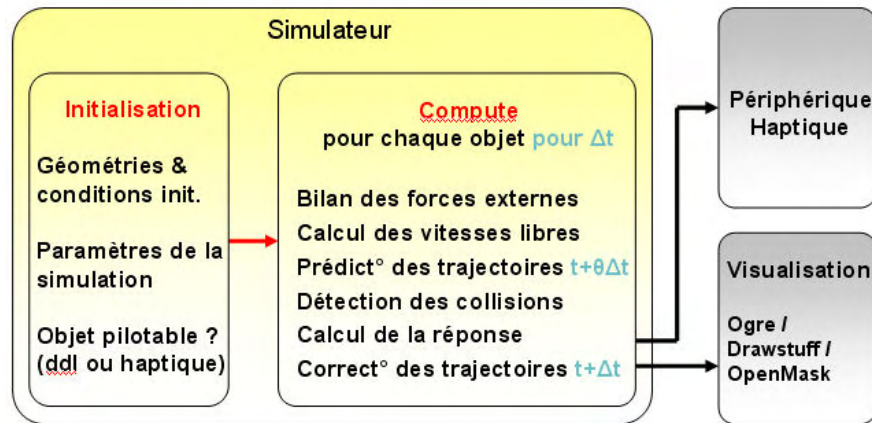


FIG. 2.7 – Architecture de Confetis.

À chaque pas de temps, Confetis se charge de traiter chaque contact de chaque objet de la scène, selon la séquence (voir figure 2.7) :

1. dans un premier temps, Confetis calcule les mouvements libres de tous les objets de la scène, par intégration classique des équations différentielles de mouvement, sans considérer les contacts entre objets. Pour cela, le simulateur réalise préalablement le bilan des efforts externes (et/ou internes), tels que la gravité, qui sont appliqués au système. Les informations d'effort (ou de position, selon si on considère un mode de couplage impédant), sont fournies par l'interface haptique et considérées comme des contraintes dynamiques ;
2. Confetis utilise alors une thêta-méthode, qui suit un schéma implicite pour prédire les trajectoires et vitesses des objets pour le demi pas de temps suivant. Renouf décrit complètement ce type d'approche dans [Ren04] ;
3. en fonction des approximations faites sur ces futurs couples position/vitesse, Confetis effectue une détection de collision entre les objets. Pour cela il applique une méthode analytique de détection des contacts, donc non optimisée, pour récupérer les configurations spatiales prédictives. Les informations calculées sont la localisation du point de contact, l'interpénétration entre les objets et la normale de direction à ce contact ;
4. Confetis applique alors les contraintes données par les lois de contacts (frottement et rebond) et applique une formulation non-régulière sur ces modèles prédictifs ;

5. il résout alors le système d'équations, formulé en un système complémentaire, grâce à des solveurs numériques de type itératifs (Gauss-Seidel) ou directs (Lemke) ;
6. la résultante de contact est alors envoyée à la boucle de contrôle (fermée) du simulateur, qui se charge seul d'intégrer les couplages virtuels (ressort-amortisseur) ;
7. Conf&ttis réintroduit finalement les informations d'efforts dans le système d'équations différentielles. Les positions et les vitesses des objets sont alors mises à jour par une nouvelle intégration de ces équations.

Ce prototype fonctionnel sert de plate-forme expérimentale à l'approche en temps-contrôlé, qui nous sert particulièrement à étudier la qualité du rendu en termes :

- de vitesse ;
- de raideur ;
- de transparence et de stabilité.

2.3.2 Couplage avec une interface à retour d'effort

Afin d'intégrer les données de commandes de l'utilisateur dans la simulation interactive, on réalise manuellement un couplage entre le contrôleur de l'interface et le moteur physique. Ce couplage haptique est réalisé au sein de l'unité de commande du périphérique par un système de couplage virtuel (ressort-amortisseur), dont les coefficients sont réglés de manière non empirique. On attache un objet virtuel de la scène au proxy de l'interface haptique, par exemple l'extrémité de la chaîne cinématique d'un bras de robot, dont les paramètres inertiels sont donnés à l'asservissement de couplage.

Il est fondamental de s'assurer que la vitesse de rafraîchissement du simulateur est passée en paramètre dans l'algorithme de couplage du périphérique. De manière générale, on estime que plus la fréquence de calcul du simulateur s'approche de 1 KHz (fréquence de rafraîchissement du contrôleur Haption Virtuose par exemple), plus on obtiendra un couplage stable. Il faut s'assurer d'autre part, que les dimensions des positions et des efforts qui sont échangées avec l'interface sont dans des plages de fonctionnement nominales du périphérique ($[0;20\text{ N}]$ et $[0;1\text{ m}]$ pour le Virtuose). Si ce n'est pas le cas, on doit mettre en place des facteurs d'échelle ou des fonctions de saturation.

2.4 Conclusion

Nos hypothèses de recherche portant sur le découplage des problèmes liés aux assemblages contraints de ceux liés à l’exploration non contrainte de la scène, nous avons abordé la gestion des collisions au sein d’une simulation. Sans effectuer une description exhaustive des techniques de détection, nous avons insisté sur le fait que cette problématique de détection est essentiellement liée aux performances temporelles imposées par l’interaction haptique. Nous verrons, au chapitre suivant, les contraintes de précisions géométriques dans le cadre des assemblages CAO.

La description des méthodes de traitement des interactions nous a rappelé les problèmes liés à la détermination des efforts de contact entre objets. Les limites des modèles dynamiques élémentaires sont atteintes pour la résolution de systèmes non linéaires discrétisés. L’effort (inconnu) de contact peut dès lors être déterminé par deux approches. Nous avons détaillé l’approche par dynamique pénalisée ainsi que celle par dynamique non régulière. L’approche pénalisée, qui modélise l’effort par une raideur virtuelle, est largement utilisée dans le secteur haptique tant son implémentation est simple et ses résultats temporels (et de stabilité) pertinents. Nous avons alors détaillé les modèles non réguliers, étudiant les approches pilotées par les événements, qui imposent une contrainte au niveau du contact une fois que celui-ci a eu lieu. Nous avons finalement détaillé le modèle par temps-contrôlé, qui prend en compte des lois comportementales tout en assurant un écoulement continu du temps.

Nous avons enfin présenté l’implémentation de l’approche *time-stepping* au sein d’un simulateur haptique. Nous avons proposé notre plate-forme expérimentale Conf&tis. Ce prototype nous permet de valider la faisabilité du couplage entre un modèle mécanique de dynamique non régulière et l’interface haptique, et ce, en respectant les contraintes identifiées au chapitre précédent. Une évaluation de notre couplage est proposée au chapitre 4, où nous étudions les performances de nos algorithmes pour les phases exploratoires d’assemblages.

Chapitre 3

Guidage virtuel contraint pour l'assemblage d'objets CAO

3.1 Introduction

Dans le contexte industriel, les applications de RV qui ont besoin d'interfaces haptiques consistent principalement à combiner des tâches d'assemblage et de désassemblage de maquettes virtuelles, à des fins de validation ou de formation. Dans ce type d'application, l'utilisation traditionnelle de l'haptique est le pilotage d'objets 3D issus de la CAO, qui peuvent interagir avec les autres objets qui constituent l'environnement virtuel.

Pour assister l'utilisateur dans ses tâches d'assemblage, les domaines de la téléopération et de la RV proposent des méthodes d'aide et de guidage dérivées de l'*assembly planning* [DCRS97]. L'objectif de ces méthodes de planification d'assemblage est de déterminer les séquences d'assemblage des différentes parties d'un produit en s'assurant que les objets ne s'interpénètrent pas dans l'environnement de travail. Cependant, dans les cas d'assemblages complexes, l'expérience de l'utilisateur peut résoudre divers problèmes de trajectoire et permet de trouver rapidement des solutions. L'expertise de l'utilisateur peut être extraite d'expériences d'assemblage et utilisée pour la génération automatique de séquences d'assemblage interactif tel que le décrit [Yua02]. Les capacités cognitives de l'utilisateur (perception de l'espace, mémoire du travail, planification etc.), permettent ainsi de produire des mouvements et séquences d'assemblage plus performants que ceux trouvés par les algorithmes de planification. Dans le cadre des assemblages interactifs, s'ils sont réalisés avec des outils multimodaux tels que les interfaces haptiques, il est alors possible d'introduire des éléments de guidage afin d'assister l'utilisateur, en lui laissant un contrôle partiel de ses mouvements. Les applications qui nécessitent à la fois une telle liberté de manipulation et un système d'assistance se trouvent essentiellement dans la formation, la validation d'assem-

blage ainsi que l'aide à la compréhension des montages si ceux-ci :

- ne proposent pas d'information *a priori* sur l'assemblage final ;
- sont composés de mouvements complexes ou d'enchaînements non intuitifs ;
- proposent des compositions d'objets non intuitives.

Dans ce chapitre, nous présentons une méthode qui combine l'utilisation de contraintes mécaniques et de *virtual fixtures*, informations abstraites ajoutées à la simulation pour la réalisation assistée de tâches d'assemblage par interface haptique. Cette méthode, nommée guidage virtuel contraint ou VCG (pour *virtual constraint guidance*), intervient dans l'espace d'assemblage tel que nous l'avons introduit à la section 1.4. Elle gère la transition entre la phase d'exploration gérée par dynamique non contrainte et la phase d'assemblage, que nous proposons de gérer par dynamique contrainte. Pour cela, notre méthode propose deux principales contributions. La première propose une nouvelle technique de guidage géométrique pour assister l'approche des objets à assembler, et plus particulièrement leur mise en position relative. La seconde porte sur le changement de mode de contrôle haptique pour contraindre ces objets à réaliser l'assemblage final. La méthode VCG applique localement une contrainte cinématique qui oblige les objets manipulés à se déplacer selon des mouvements et trajectoires prédéterminées. L'avantage mis en avant par cette méthode est le découplage complet de la phase exploratoire portée par une dynamique haptique classique, de la phase d'assemblage où, grâce à cette méthode, il est possible de désactiver la détection de collision entre des objets directement issus de la CAO.

Ce chapitre est organisé comme suit : la section 3.2 présente, en contexte, un état de l'art sur les assemblages contraints d'objets issus de la CAO, où nous mettons en évidence les problèmes liés aux géométries des objets issus de la CAO et de leur exploitation dans les simulations de RV. Nous y étudions également les limites des détecteurs de collision entre objets 3D dans le cadre précis des simulations d'assemblage. La section 3.3 introduit le concept de guidage virtuel contraint, notre méthode générique d'aide à l'assemblage d'objets CAO en environnement RV. Nous détaillons l'utilisation combinée de guides géométriques et de contraintes cinématiques au cœur de la simulation interactive pour la réalisation de tâches d'assemblage. Dans la section 3.3.2, nous proposons une nouvelle méthode de positionnement haptique d'objets 3D dont nous détaillons les stratégies de guidage géométrique. La section 3.3.7 présente les détails du changement de mode de contrôle dynamique à la transition entre la phase exploratoire et la phase d'assemblage. Nous présentons enfin, section 3.4, nos conclusions et les travaux à réaliser pour étendre l'utilisation de la méthode VCG aux assemblages complexes.

3.2 Etat de l'art des assemblages d'objets CAO en environnement RV

Les modèles numériques issus des différentes sources peuvent toutefois induire des problèmes de compatibilité, interdisant ainsi l'utilisation de séquences d'assemblage simples si les données 3D ne sont pas modifiées lors de leur utilisation sur des plates-formes externes [BS96]. Tandis que la conception d'un produit peut nécessiter la réunion de données 3D qui ont été créées par des techniques ou des solutions logicielles différentes, les assemblages interactifs sont, quant à eux, simulés sur des plates-formes de réalité virtuelle permettant les interactions haptiques entre les utilisateurs et les objets 3D. La principale limite de cette forme de conception émerge donc à la croisée des deux domaines, indépendamment des formats d'échange de données 3D entre la CAO et la RV, qui sont à eux seuls un vaste champ de recherche. En outre, si elles disposent de données d'assemblage exploitables (surfaces en contact, types de liaison entre objets etc.), peu de plates-formes de CAO proposent des protocoles d'interaction, au sens de la réalité virtuelle, pour simuler directement les assemblages. Les difficultés liées à l'exploitation de ces données d'assemblage résident dans leur réutilisation dans les simulations haptiques, qui s'emploient principalement à donner des comportements physiques aux géométries en collision.

Dans la section suivante, nous décrivons les difficultés liées à l'assemblage haptique d'objets issus de CAO dans les simulations de RV. Nous étudions d'abord les limites de l'exploitation des objets CAO. Pour cela, nous soulignons les problèmes liés à l'utilisation simultanée des géométries et des données d'assemblage issus des logiciels de CAO, dans des simulations interactives externes. Nous étudions ensuite les problèmes inhérents à l'exploitation, dans les simulateurs RV, des géométries directement issues des modeleurs CAO. Pour cela, nous mettons en exergue les limites des méthodes de détection des collisions pour les assemblages d'objets issus de CAO en soulignant notamment les erreurs géométriques qui leur sont appliquées. Nous étudions enfin les méthodes existantes qui permettent l'assemblage interactif d'objets directement dans les modeleurs CAO, ainsi que les méthodes de guidage mises en place pour assister l'utilisateur dans la réalisation de l'assemblage.

3.2.1 Exploitation des objets issus de CAO

L'exploitation des objets CAO dans des simulations de RV est fondamentalement limitée par l'échange des données entre les applications. En effet, au moment de la "conception CAO" d'objets 3D, ceux-ci sont représentés par des modèles contenant des informations analytiques, topologiques et géométriques. Ces informations sont, par exemple, nécessaires pour réaliser la mise en production

ou l'analyse mécanique du produit (calcul de structure, déformation, dissipation énergétique etc.). Toutefois, seules les données géométriques sont basiquement requises pour l'exploitation de ces objets dans les simulations de RV. La plupart des autres informations contenues dans les modèles CAO, telles que les paramètres d'assemblage (mais aussi matériaux, masses, maillage par éléments finis etc.), sont réinjectées de manière externe et différente pour chaque application de RV.

Les modelleurs géométriques des logiciels de CAO utilisent généralement des représentations surfaciques où seules les surfaces frontières de l'objet sont modélisées. Les géométries y sont décrites par des NURBS (B-Spline rationnelle non uniforme, surface ou courbe gauche dans l'espace) selon une approche B-Rep (Boundary REPresentation) [HHJS99]. Dans les applications de type RV, les objets 3D sont principalement représentés par des modèles polygonaux, généralement des approximations polyédriques des surfaces paramétriques des objets issus des modelleurs CAO. Ce type de représentation par facettes planes est communément utilisé en raison des performances accrues des algorithmes de rendus (graphiques) portés sur les cartes graphiques 3D. En fonction de la complexité de la scène virtuelle, et donc du nombre de polygones à afficher, de nombreuses techniques décomposent la scène en plusieurs niveaux de détails. Les méthodes de simplification polygonale qui sont alors appliquées, utilisent des niveaux de détail (LOD) et des distances objets-observateur pour décomposer la scène [Kru99].

Les géométries facettisées peuvent restreindre, à elles seules, certains mouvements relatifs entre les objets. Ces restrictions sont dues à l'apparition de contacts entre les surfaces maillées, qui n'existent pas si on considère les surfaces théoriques des objets. En effet, l'introduction de ces contacts peut limiter le nombre de DDL entre les objets maillés au moment de l'assemblage, surtout si celui-ci n'a pas pris en compte les erreurs dues au maillage. La figure 3.1 illustre, pour un exemple d'insertion, la diminution du nombre de DDL entre un pion et un trou. Théoriquement cylindriques au niveau de la CAO, le pion et le trou (sur la figure de gauche) forment une liaison pivot-glissant (2 DDL) au moment de leur insertion. Une fois les objets facettisés (très grossièrement facettisés sur la figure de droite), le seul mouvement possible devient une translation verticale qui correspond à une liaison glissière (1 DDL) entre les objets. Le type de liaison entre les objets, qui définit leur nombre de DDL, est fondamentalement lié au nombre et au type de contact entre les objets [TKI99]. Les surfaces impliquées jouent donc un rôle déterminant dans les trajectoires des objets pour leur assemblage / désassemblage. Dans l'exemple de gauche de la figure 3.1, il suffit qu'une des surfaces du prisme interpénètre la surface du tour, pour que tous mouvements relatifs soient réduits à zéro.

Les deux différents paradigmes de représentation géométrique, de la CAO et de la RV, impliquent la nécessité de convertir les données pour les rendre com-

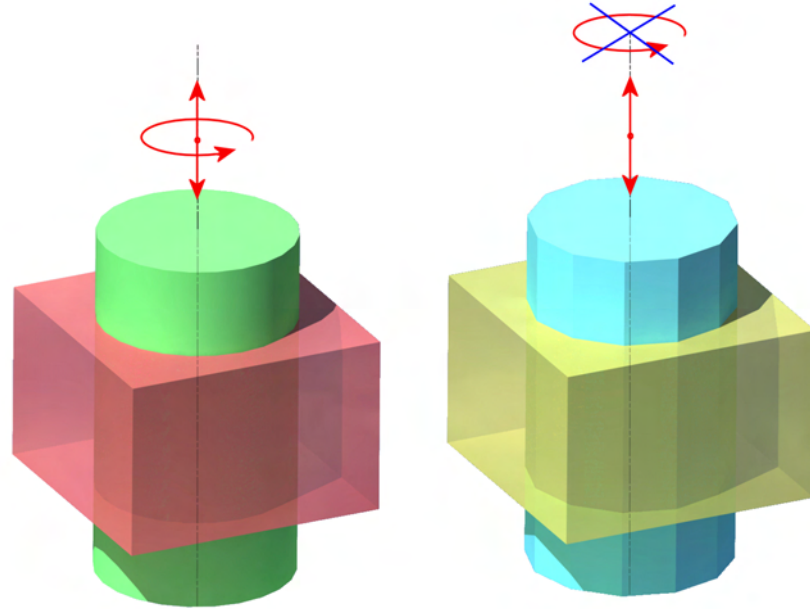


FIG. 3.1 – Représentation 3D du changement du nombre de DDL entre objets, entre les géométries théoriques (à gauche) et les géométries facettisées (à droite).

patibles entre elles. Dans le cadre de nos travaux, cela revient à convertir les données géométriques incluses dans les formats d'objets CAO, dans des formats compatibles aux simulations de RV. Dès lors, on ne parle plus d'objets CAO mais de représentation géométrique issue de la CAO, puisque seules les informations géométriques sont ré-exploitées. Ces données géométriques peuvent toutefois être associées à des informations supplémentaires (topologies, matériaux, propriétés physiques, rendu graphique etc.) si elles font partie de formats d'échange de données 3D tels que 3DXML ou COLLADA.

Ces conversions sont réalisées par les modules d'export des logiciels CAO, qui génèrent des modèles géométriques surfaciques (paramétriques). C'est ce dernier mode de représentation qui est exporté et utilisé en RV. Chaque surface du solide est décrite par un ensemble de facettes, généralement polygonales, dont la densité varie en fonction de l'erreur d'approximation. Cette procédure de conversion est standard dans l'industrie, en particulier pour le processus d'échange de données 3D. Les objets CAO y sont représentés majoritairement par :

- le format STL (STereoLithography), issu du domaine du prototypage rapide, qui facettise les surfaces de l'objet en triangles ;
- le format IGES (Initial Graphics Exchange Specification), format d'échange des modeleurs CAO, pour les surfaces NURBS.

Toutefois, les modèles facettisés obtenus peuvent posséder des défauts (normales de triangles inversées ou triangles absents) qui imposent le passage des maillages dans des logiciels spécialisés dans la reconstruction de surfaces. De même, il peut être nécessaire de convertir ces modèles plusieurs autres fois, souvent par d'autres logiciels de conversion, afin d'obtenir le format 3D requis par le simulateur RV. Pour chaque conversion du modèle original CAO, résultent des pertes de détail des objets, intrinsèques aux méthodes de facettisation. Les approximations géométriques engendrées par le changement de représentation des objets CAO, font perdre des précisions géométriques fondamentales pour leur assemblage ultérieur.

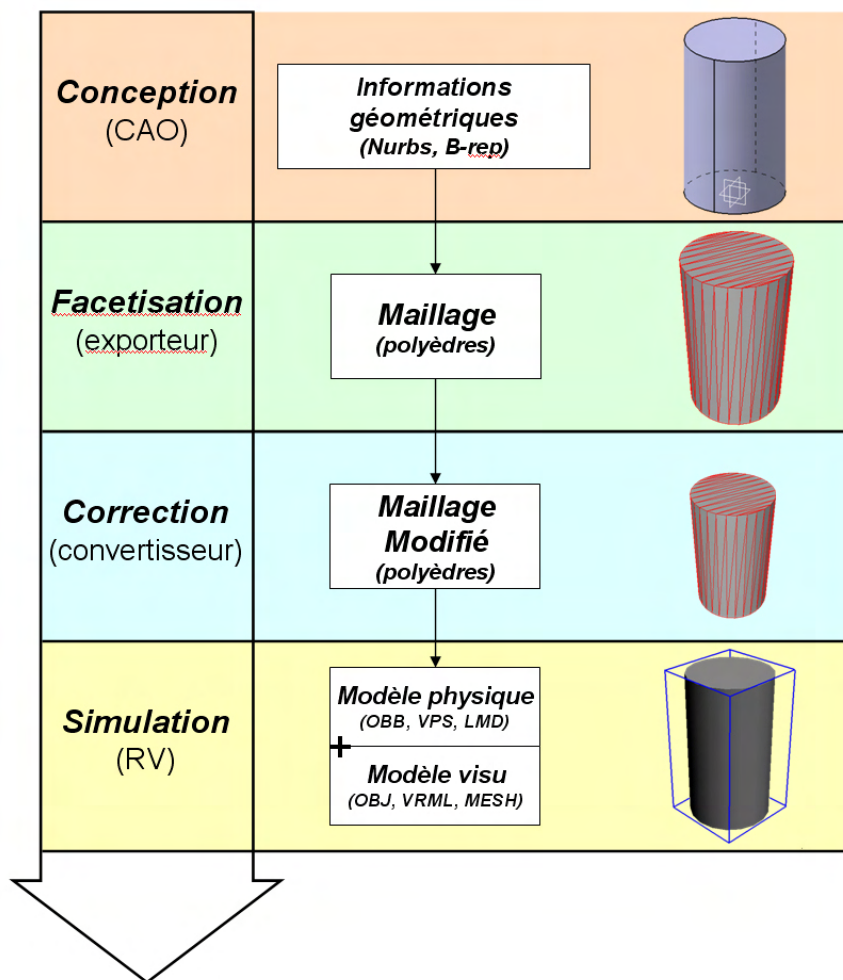


FIG. 3.2 – Processus classique de traitement des objets CAO, pour leur utilisation en simulation de réalité virtuelle

La figure 3.2 illustre un processus classique de traitement d'un objet CAO, ici

un cylindre, pour son exploitation dans une simulation de RV. Dans cet exemple, le cylindre original est converti trois fois avant de pouvoir être utilisable, au sens de l'assemblage, sur une plate-forme de RV externe :

1. les géométries de l'objet CAO sont premièrement facettisées puis exportées dans un format polygonalisé de type STL. Cette étape correspond à la transition entre les zones orange et verte de la figure 3.2. Notons qu'en fonction du type de données géométriques, exploitables en sortie de modelleur CAO, il peut être nécessaire de rajouter une étape de conversion intermédiaire ;
2. la géométrie est alors convertie en un format géométrique compatible avec la simulation. Elle peut également être modifiée en prévision des modélisations réalisées par les algorithmes de détection de collision (voir section 3.2.2). Cette étape correspond à la transition entre les zones verte et bleue de la figure 3.2 ;
3. la géométrie est alors importée dans la simulation interactive où elle est d'une part dupliquée puis modifiée pour la rendre compatible avec les moteurs physiques (physicalisation), et d'autre part utilisée par les moteurs de rendu graphique. Là encore elle pourra être dupliquée et re-facettisée à différentes échelles (multi représentation). Cette étape correspond à la transition entre les zones bleue et jaune de la figure 3.2.

Dans la suite de ce chapitre, nous illustrons nos propos avec l'exemple connu sous le nom de *peg-in-hole*. Cet assemblage consiste à insérer un pion prismatique, contrôlé par l'utilisateur via l'interface haptique, dans un trou de forme conjuguée situé dans un objet fixe de la scène. Cet exemple sert de référence à de nombreuses études en robotique et en haptique. D'une part cet exemple fait intervenir des objets très simples qui sont aisément manipulables par les utilisateurs. D'autre part les problèmes rencontrés dans cet exemple (trajectoires d'approche et mouvements d'insertion, phénomènes de frottement et coincement, gestion de multiples contacts conformants etc.) représentent la plupart des problèmes rencontrés dans les simulations physiques. Le type de tâche canonique qu'il fait intervenir a été largement étudié dans les domaines de la téléopération et de la simulation interactive. Lim *et al.* [LRD⁺07] proposent l'étude complète d'un système d'assemblage assisté par ordinateur, évaluant les performances des utilisateurs pour des tâches d'assemblage haptique. Ils réalisent leurs expériences sur le *peg-in-hole* afin d'étudier des facteurs tels que la stéréoscopie ou l'utilisation de chanfreins sur les modèles, sur la productivité.

La figure 3.3 illustre, en vue de coupe, l'incompatibilité d'assemblage induite par les conversions de données. Sur la partie gauche de la figure, les deux objets sont créés directement dans le modelleur CAO et sont représentés comme étant assemblés avec une parfaite complémentarité des géométries qui les représente.

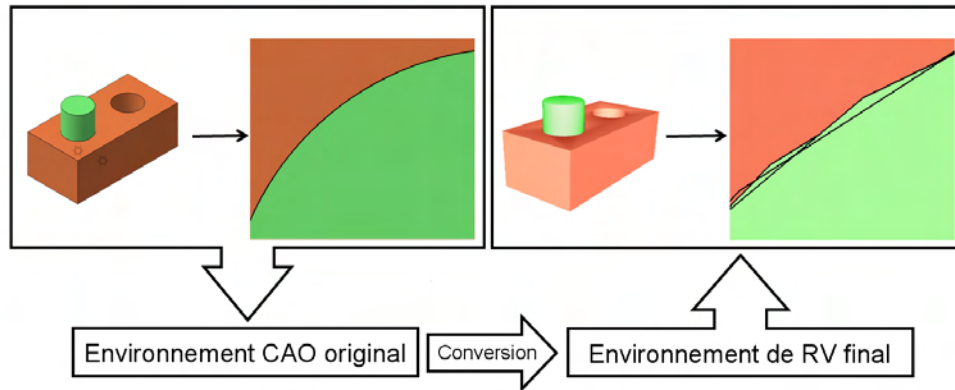


FIG. 3.3 – Perte de précision géométrique lors du changement de représentation de la CAO vers la RV, sur un exemple de *peg-in-hole*

Le pion est ainsi représenté par un prisme à base circulaire, associé à une information topologique de cylindricité, tandis que le trou est créé par extrusion dans un volume, à partir des données du pion précédent. Disposant ainsi des mêmes données géométriques (diamètre et/ou profondeur), ils sont créés et affichés simultanément dans l'environnement CAO de sorte que leur assemblage est théoriquement parfait. Sur cette vue de dessus, la correspondance des arcs de cercle signifie la cylindricité et concentricité idéales des deux objets. Une fois que ces deux objets ont subi des facettisations, c'est-à-dire qu'ils sont représentés par des surfaces maillées, la correspondance géométrique n'est plus assurée. Sur la partie droite de la figure, si on superpose les objets tels qu'ils l'étaient dans la CAO (avec le même repère d'assemblage) on constate immédiatement que les objets voient certaines de leurs facettes s'interpénétrer, rendant impossible, en termes de détection de collision, leur assemblage au sein d'une simulation de RV.

Des solutions existent pour pallier ce problème d'exploitation. Ces solutions s'emploient à rendre compatibles les données 3D issues de plusieurs plates-formes, mais il n'existe pas de modèle générique salubre permettant d'automatiser le passage de données CAO dans un format d'échange avec la RV. Pour pallier le problème de changement de représentation et de transfert inter plates-formes, des solutions telles que VADE [JWJ⁺99] permettent d'échanger directement des géométries et des informations de trajectoires avec les environnements CAO. Elles permettent également de calculer des mouvements et contraintes cinématiques pour assister l'assemblage virtuel. Bourdot *et al.* [BCP⁺08] intègrent RV et CAO dans un *framework* commun qui rend possible l'édition implicite et intuitive d'objets 3D au sein d'un environnement virtuel. Seth, dans ses travaux de thèse [Set07], propose un état de l'art complet des simulations d'assemblages CAO en environnement virtuel. Une limite majeure à ce genre de solutions duales

réside, toutefois, dans le fait qu'elles ne permettent pas l'exploitation générique des géométries issues de la CAO dans des environnements de réalité virtuelle. Ces solutions ne consistent, en effet, qu'à l'utilisation de plates-formes communes à la CAO et la RV en proposant un certain nombre d'outils de guidage efficaces, mais uniquement exploitables sur ces plates-formes.

3.2.2 Limites de la détection des collisions dans les simulations d'assemblage

Les simulations physiques, qu'elles autorisent l'interaction haptique ou non, détectent les configurations spatio-temporelles des collisions entre les objets simulés. Dans le but de déterminer les efforts de réaction induits par ces collisions, et ainsi de les proposer à l'utilisateur via l'interface haptique, les simulations procèdent à une étape de détection des contacts en temps-réel dans l'environnement virtuel. Dans les simulations haptiques, la plupart des méthodes de détection des collisions décomposent le processus de détection en plusieurs phases, les *narrow phases* et les *broad phases*, qui sont liées à la proximité des objets candidats à la collision (voir chapitre 2.2.1). Dans ces deux phases, les algorithmes changent les représentations des objets, en recréant des géométries sur lesquelles sont effectués les calculs de collision. Si les géométries facettisées sont préservées pour la visualisation, elles sont dupliquées et transformées (généralement simplifiées) pour :

- minimiser le temps de calcul alloué aux détections, accélérant ainsi la phase de simulation physique ;
- les rendre compatibles avec certaines méthodes de traitement physique.

La figure 3.4 illustre deux méthodes de détection de collision qui détectent la position du point local de collision entre deux engrenages en contact. Les deux méthodes, basée à gauche sur des calculs de pénétration de volumes englobants [GLM96] et à droite sur le calcul de distance inter-polygone [JW03], utilisent des géométries simplifiées dérivées des maillages entrés dans le simulateur physique, dont les surfaces sont représentées en noir sur la figure.

Les méthodes LMD et VPS que nous avons présenté à la section 2.2.1, utilisent des maillages spécifiques pour détecter des proximités ou pénétrations inter-objets. Qu'il s'agisse de volumes englobants ou de restrictions des géométries en nuage de points (cas de VPS par exemple), les calculs physiques d'interaction sont réalisés sur ces géométries. Les erreurs induites par ces nouvelles transformations, alors même que les géométries des objets 3D en entrée de simulateur sont elles-mêmes des approximations des géométries CAO théoriques, suffisent à interdire les assemblages d'objets dont les cotes théoriques sont ajustées. Notons que dans le processus même de "conception CAO", ces ajustements ne sont

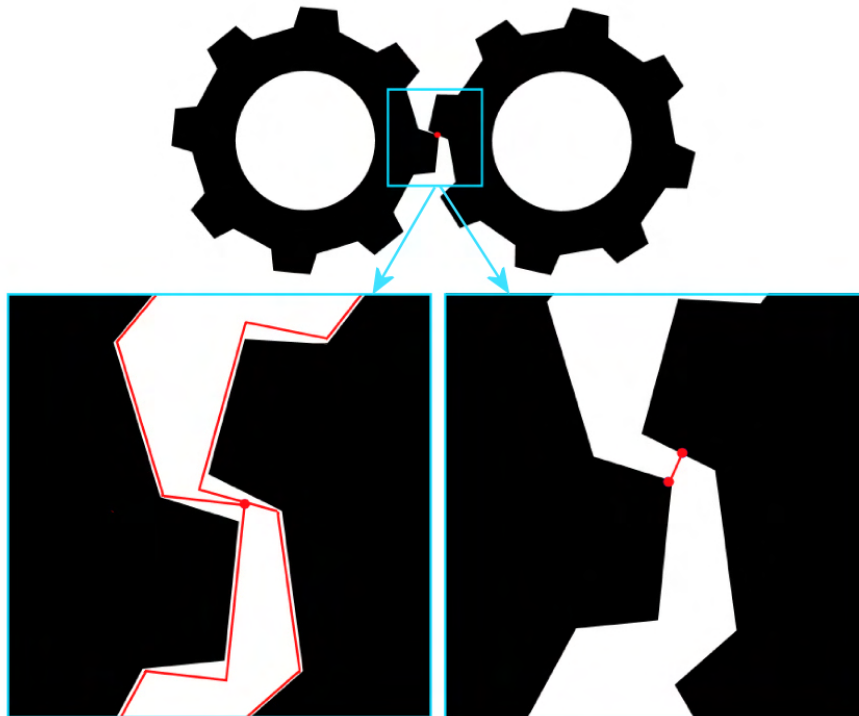


FIG. 3.4 – Exemple de localisation de point de contact entre deux roues dentées discrétisées. À gauche le contact est localisé à l'intersection de volumes englobants (boîtes rouges). À droite, le contact est défini comme une distance seuil entre géométries élémentaires.

pas représentables en 3D (ni même 2D), mais signifiés par des cotations mécaniques normalisées ; dans ces cas réels, il incombe à la production de respecter les tolérances de fabrication pour assurer la montabilité des pièces.

D'autres algorithmes de détection de collision, tel que Vortex de CMLabs qui utilise une méthode GJK (Gilbert-Johnson-Keerthi), autorisent l'interpénétration entre les objets pour calculer les efforts de collisions. Ces classes de méthodes fonctionnent très bien pour l'exploration de scènes haptiques qui font intervenir des contacts unilatéraux. Toutefois, en autorisant l'interpénétration des géométries, elles rendent impossibles les assemblages qui font intervenir des contacts bilatéraux.

En pratique, pour réaliser des assemblages précis d'objets issus de la CAO, il faut s'assurer que les géométries utilisées par les algorithmes de collision respectent les données géométriques des assemblages. Ces données sont décrites dans les modelleurs CAO au moment de la création des objets, mais ne correspondent qu'à des données théoriques, généralement prévues pour la production des produits. Pour s'assurer qu'elles sont respectées dans le simulateur, il faut prendre

en compte, rétroactivement, les approximations géométriques créées par les algorithmes de collision. Cela signifie qu'il faut modifier les géométries des objets avant leur introduction dans les moteurs de collision. Sans maîtrise du processus de maillage des algorithmes de détection, ces modifications sont réalisées itérativement de sorte que le l'assemblage final soit satisfaisant. Toutefois, la modification des formes ou des dimensions des objets, qu'elle soit réalisée directement sur le modèle CAO ou sur son maillage, implique des sacrifices. Les approximations engendrées influent sur la précision des géométries, par rapport à leurs versions "originales" CAO. Cela est directement répercuté sur les précisions des assemblages, dont on vient de dire qu'elles sont déjà, à la base, inatteignables.

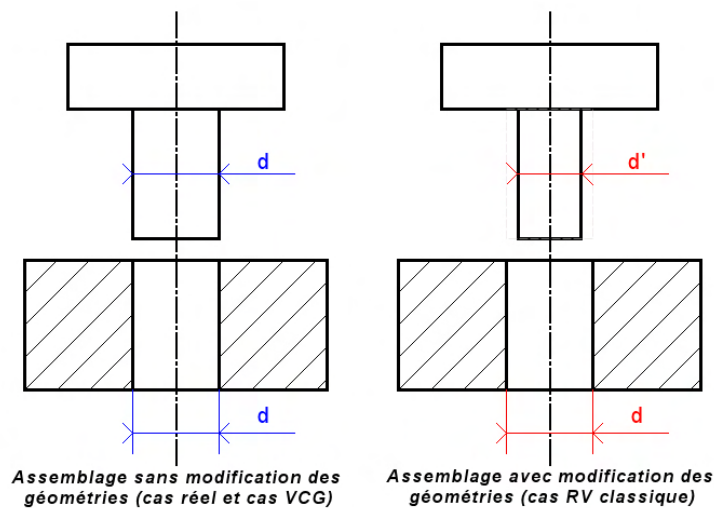


FIG. 3.5 – Dessin technique 2D du montage *peg-in-hole*, sans et avec changement de diamètre du pion

Dans l'exemple du *peg-in-hole*, illustré figure 3.5, il est nécessaire de diminuer le diamètre du pion (ou du trou) afin de pouvoir insérer les deux objets. À droite, le pion et le trou sont conçus "normalement" sans différentiation de leurs diamètres respectifs d . À gauche, le diamètre d' du pion doit être modifié ($d' < d$) pour assurer le montage ultérieur en simulation RV.

Indépendamment des compatibilités de formes évoquées précédemment, ce changement de diamètre doit être au minimum égal au seuil de tolérance de la méthode de détection. La figure 3.6 illustre, d'un point de vue purement dimensionnel, les modifications géométriques du pion, pour son utilisation avec un volume englobant. Une des façons de les modifier, *a posteriori* de sa conception CAO, rappelons-le, est de modifier son diamètre d_1 et sa longueur l_1 . Son nouveau diamètre d_2 et sa nouvelle longueur l_2 doivent alors être paramétrés de sorte que, une fois le volume englobant en place (ici un volume cylindrique), les dimensions

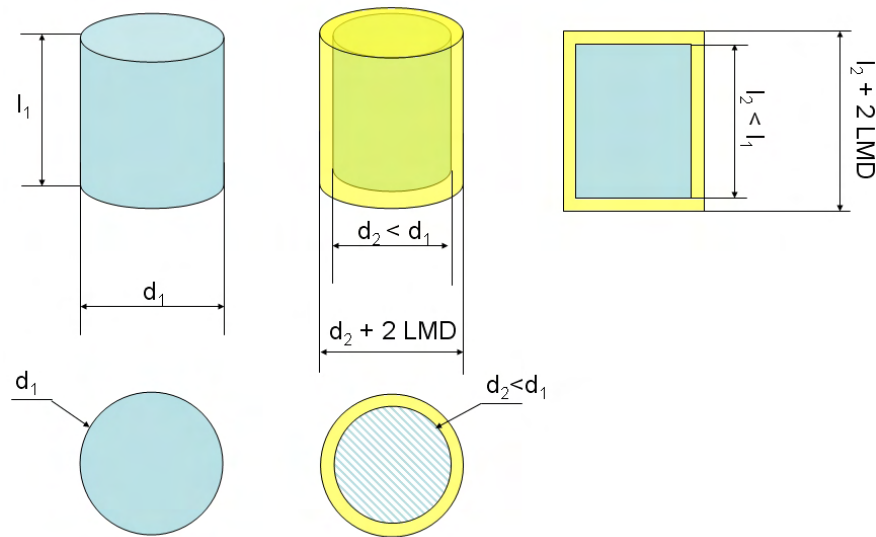


FIG. 3.6 – Conditions de modification des géométries du pion, pour l'utilisation dans un algorithme de détection de type *LMD*.

maximales de celui-ci soient inférieures ou égales à celles de l'objet initial. Bien sûr cet exemple n'a pas de finalité réaliste puisqu'en pratique il faut modifier de même les géométries du trou, et puisque les techniques de détection par volumes englobants existantes ne requièrent pas de tels écarts géométriques. Toutefois, on soulignera que cette démarche est globalement utilisée pour réaliser rapidement des assemblages haptiques.

La modification d'une géométrie paraît intuitivement simple pour l'exemple du *peg-in-hole*, où il suffit de modifier en CAO la valeur d'un paramètre. Cependant, il n'est pas trivial de pouvoir modifier une cote géométrique précise sur un objet complexe, surtout s'il fait lui-même partie d'un assemblage. Si l'arborescence de conception ne permet pas de modifier cette cote (héritages complexes), l'alternative réside en la modification "brutale" du maillage, telle que la mise à l'échelle globale de l'objet, avec toutes les approximations géométriques que cela implique. Cette dernière approche interdit d'ailleurs toute combinaison ou enchaînement d'assemblage.

La figure 3.7 illustre l'exemple du *peg-in-hole* au sein d'une simulation interactive. Sur cette figure, deux pions sont positionnés au-dessus de deux trous identiques. À gauche, le diamètre du pion bleu est le même que celui du trou au-dessus duquel il est positionné. Ces deux objets ont été créés ensemble dans le modelleur CAO Dassault Système CATIA, avec exactement les mêmes cotes. Pour être exploitables sur la plate-forme de RV, les objets CATIA ont été maillés et exportés dans un format STL. Puis ils ont été convertis en format OBJ par un

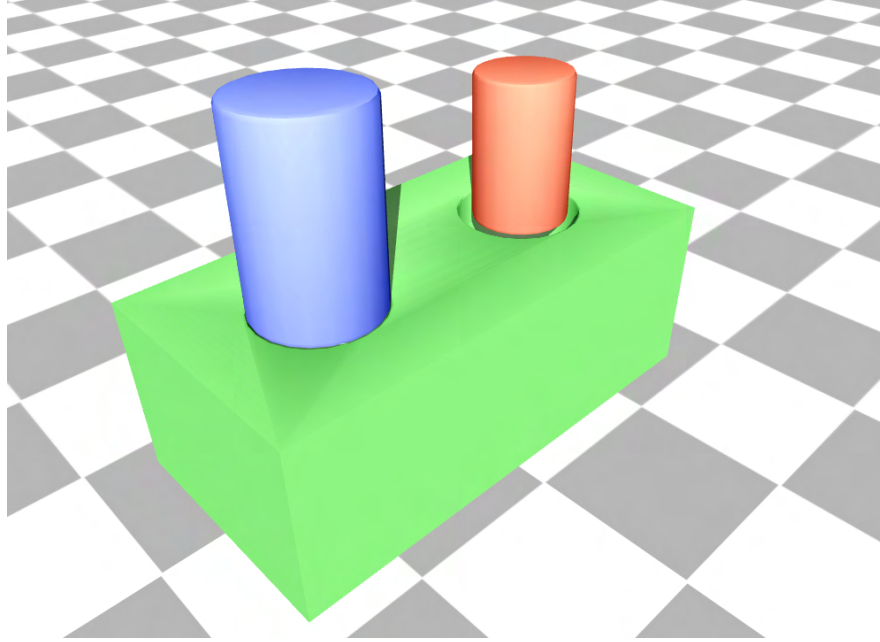


FIG. 3.7 – Exemple *peg-in-hole* où le diamètre du pion est non ajusté (à droite) et ajusté (à gauche), au sein de la simulation de RV.

logiciel externe de conversion. Une fois importés dans la simulation, les maillages ont enfin été modifiés par la méthode LMD (voir chapitre 2.2.4). C'est sur ces maillages finaux, non visibles, qu'ont lieu tous les calculs d'interaction. Toutefois leur assemblage est impossible en l'état : d'une part car la nature même des géométries discrétisées les empêchent de trouver une quelconque correspondance surfacique, et d'autre part car la méthode LMD considère un "maillage de calcul" fondamentalement différent de l'objet CAO original. Aussi pour pouvoir réaliser cette insertion *peg-in-hole*, nous avons modifié significativement (par itérations), le diamètre du pion jusqu'à ce qu'il puisse être assemblé dans le trou (pièce rouge sur la figure 3.7).

3.2.3 La métaphore des guides virtuels

Rosenberg propose un moyen permettant d'améliorer les performances d'un opérateur humain lors des tâches de télémanipulation. Il pose, dans le cadre de la téléprésence, le concept de guides virtuels (*virtual fixtures*), qui améliore jusqu'à 70 % les performances d'un opérateur des tâches d'insertion. Introduites dans [Ros92] et [Ros93], les *virtual fixtures* sont des informations perceptuelles abstraites ajoutées à la simulation interactive pour aider les utilisateurs à réaliser des tâches spécifiques. Ces classes de modes de guidage, qui sont implémentées

à haut niveau dans les logiciels, limitent les mouvements des utilisateurs, leur interdisent l'accès à certains espaces ou influencent leurs mouvements selon des trajectoires désirées. Les guides virtuels sont des métaphores qui consistent en l'ajout de volumes et surfaces virtuels qui sont perçues par l'utilisateur selon plusieurs modalités (visuelle, auditive et haptique).

Otmane [Ota00] propose plusieurs méthodes de construction des guides virtuels afin de permettre à l'utilisateur d'accomplir un grand nombre de tâches. Il propose des méthodes de création de guides selon :

- différents types de surfaces (plan, disque, etc.), qui limitent les zones accessibles de l'environnement ;
- des volumes ouverts (tubes), extensions volumiques de ces surfaces, qui forcent l'objet à se déplacer dans des zones ciblées de l'espace ;
- des volumes fermés (sphère, cône, superconiques et superquadriques) qui peuvent être traversés par l'objet, et à l'intérieur desquels on lui applique des champs de force répulsifs ou attractifs ;
- des courbes (BSplines) qui peuvent servir de trajectoires à l'objet.

Kheddar [Khe97] définit une structure globale pour les guides virtuels à l'aide d'un ensemble de champs optionnels. Nous adoptons ce formalisme dans notre méthode de guidage géométrique. Selon la nature et le contexte d'utilisation du guide, on définit systématiquement :

- l'attachement : le guide virtuel peut être attaché à un objet virtuel ou à un endroit de l'environnement virtuel. Il peut être attaché d'une manière statique (il est alors figé à un endroit ou attaché à un objet virtuel particulier) ou dynamique (il apparaît suite à un événement, par exemple, lors de la détection de collisions). Ainsi, on définit pour chaque guide une position et une orientation dans l'espace défini par l'environnement virtuel ;
- la zone d'influence : le guide virtuel peut être associé à une zone (de forme volumique, surfacique ou autre) qui jouera le rôle d'un bassin d'attraction ou plus simplement d'une zone d'action. Ces zones d'influence sont définies par des équations analytiques (statique ou paramétrée) qui délimitent (totalement ou partiellement) la forme du guide ;
- la condition d'activation : le guide est associé à une condition d'activation. Elle peut se traduire par l'appartenance d'un ensemble de paramètres extérieurs à la zone d'influence ou par une autre condition événementielle ;
- la condition de désactivation : la condition de désactivation rend inactive la fonction du guide virtuel. Elle peut être définie comme une négation de la condition d'activation ou l'atteinte d'un état final désiré.

Dans le cadre de l'assemblage interactif, nous utilisons ce formalisme pour positionner des objets dans l'espace. Nous souhaitons, en effet, assister l'utilisateur

en contraignant les zones accessibles de l'espace, tout en le laissant libre de toute perturbation extérieure. Nous conservons la fonction de *murs virtuels* proposée par Rosenberg, qui diffère de l'approche par magnétisation, telle qu'elle est proposée par Payandeh [PS02] ou Kuang *et al.* [KPZ⁺04]. Afin de positionner l'objet dans l'espace, en le forçant à suivre des directions (ou positions) particulières, les auteurs appliquent des champs de forces orientées (champs magnétiques virtuels) lorsqu'il entre en collision avec des volumes virtuels élémentaires disposés sur la scène.

3.2.4 Conclusion sur les assemblages d'objets CAO en environnement virtuel

Les approximations géométriques dues aux facettisations, combinées aux erreurs géométriques induites par la détection de collision, rendent impossible l'assemblage direct d'objets CAO, sans en changer préalablement leurs géométries (par simplification ou mise à l'échelle). Pour permettre leur exploitation, les différents changements de représentation effectués sur les objets CAO introduisent des erreurs géométriques qui interdisent leurs assemblages directs dans les simulations externes. Le processus de facettisation, qui peut être répété plusieurs fois entre la CAO et la RV, effectue des approximations sur les géométries qui peuvent rendre impossible certaines tâches d'assemblage haptique. Parallèlement, les algorithmes de détection utilisés dans ces simulations haptiques maillent, voire discrétisent, de nouveau les représentations géométriques, de sorte que les erreurs introduites empêchent d'autant plus les assemblages.

Ces deux facteurs, montrent les limites de l'exploitation des objets issus de la CAO vers des simulations interactives de RV. Le prix des changements de représentation consécutifs, additionné aux imprécisions des algorithmes de collision, contraint les créateurs d'applications à modifier fortement les objets virtuels avant de pouvoir les utiliser dans les simulations d'assemblage.

L'emploi réel des objets CAO pour l'assemblage haptique réside dans l'exportation des modèles CAO et la conversion dans des modèles de RV compatibles avec une modification notoire des géométries (dimensions, formes, zone d'intérêt) pour des scénarii d'assemblage très précis.

La méthode que nous proposons dans la section suivante, contourne les problèmes de précision de maillage et de détection de collision, pour l'assemblage direct des objets en sortie d'exporteur CAO. En considérant les propriétés d'assemblage directement à partir des modeleurs CAO, cette méthode propose donc l'assistance de l'utilisateur pour l'assemblage d'objets 3D (dont ceux bruts de CAO) dans des environnements interactifs "externes".

Dans la suite de ce mémoire, nous nous permettrons de nommer *objets CAO* les modèles géométriques issus des modeleurs CAO qui ont été rendus compatibles

avec les plates-formes de RV. Ces modèles géométriques ont, par exemple, été convertis en maillages triangulaires sans transformation majeure de leurs formes ou de leurs dimensions.

3.3 Contribution : guidage virtuel contraint pour l'assemblage d'objets CAO

Nous proposons une approche générique pour l'assemblage d'objets issus de la CAO dans des environnements de RV et introduisons le concept de guidage virtuel contraint, également appelé VCG pour *virtual constraint guidance*. Le concept majeur de notre méthode tient dans la combinaison de *virtual fixtures* avec des contraintes mécaniques qui limitent le nombre de degrés de liberté durant les tâches d'assemblage. Dans le reste de ce mémoire nous garderons l'appellation de *virtual fixtures* telles qu'elles sont définies par Rosenberg, plutôt que d'utiliser une traduction francophone (*guides virtuels*) moins précise et généralement réductrice.

L'innovation de notre approche tient dans le fait que nous proposons d'utiliser les géométries directement issues des modeleurs CAO, sans modifier leur forme ou les dimensions de leurs maillages, pour les assembler interactivement. Ces objets sont donc sujet aux lois physiques utilisées en haptique (voir section 2.2.5), à la différence que ces lois diffèrent entre la phase exploratoire et la phase fonctionnelle telles que nous les avons introduites à la section 1.4). Dans cette partie, nous ajoutons une décomposition supplémentaire au sein de la phase fonctionnelle. Nous séparons la *phase d'approche* de l'objet vers sa destination (*i.e.* l'endroit où il doit être assemblé), de la *phase d'assemblage* où la finalisation de l'assemblage est réalisée grâce à une contrainte. En effet, lors de cette dernière tâche, nous proposons de désactiver le mode de traitement par détection de collision au profit d'un contrôle par contraintes d'assemblage. Pour cela, nous considérons dans un premier temps qu'une tâche d'assemblage peut être décomposée comme la somme de tâches élémentaires qui sont modélisables par des liaisons mécaniques simples (glissière, pivot, appui plan...).

Dans ce chapitre, nous illustrerons nos arguments autour de l'exemple de l'insertion d'un pion dans un trou aux formes conjuguées. Cet exemple, aussi appelé *peg-in-hole* est bien connu du domaine de l'haptique et est largement traité dans la littérature. Nous considérons ainsi que le mouvement d'insertion d'un pion (cylindre) dans un trou de même diamètre, voir figure 3.8, peut basiquement être modélisé en une liaison pivot-glissant entre les deux objets, introduisant ainsi une contrainte bilatérale dans le système physique.

Dans ce chapitre, nous présentons notre méthode de guidage sur des exemples simples d'assemblage, et particulièrement sur des tâches d'insertion. Dans nos exemples, nous considérons basiquement que les utilisateurs ont une connaissance

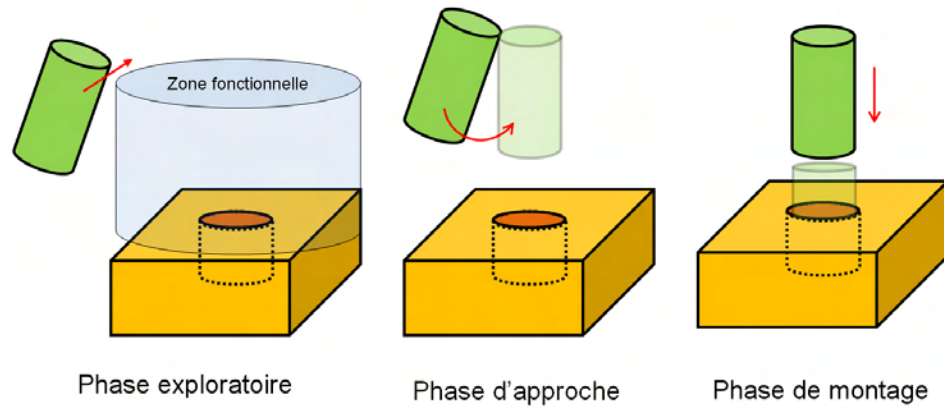


FIG. 3.8 – Décomposition du guidage. À gauche, l'objet est en phase exploratoire, aucun guidage n'est activé. Au milieu, l'objet est en phase d'approche, guidé par un guidage géométrique (non représenté). À droite, l'objet est en phase d'assemblage, guidé par un guidage contraint (non représenté).

existante des assemblages à réaliser et qu'ils manipulent "convenablement" les objets selon les scénarii proposés. Nous remettons ces hypothèses en cause lors de la présentation des expériences, développées au chapitre 4. Nous discutons également de l'extension de notre méthode pour des cas d'assemblages complexes, c'est-à-dire :

- des assemblages de plusieurs niveaux, qui font intervenir des montages (ou démontages) successifs sur les mêmes objets ;
- des assemblages imposent des trajectoires de montage complexes ;
- des assemblages composés d'objets déformables (flexibles, joints, etc.).

3.3.1 Concept : considérer l'assemblage comme une combinaison de liaisons mécaniques

Comme nous le proposons à la section 1.4, nous dissocions l'espace d'exploration de l'espace d'assemblage fonctionnel. Dans le premier espace, l'utilisateur manipule un objet selon tous les DDL disponibles (relatifs à l'interface) et le fait interagir naturellement avec les autres objets de la scène virtuelle. Dans le second espace, qui est relatif à la tâche d'assemblage, nous proposons d'assister l'utilisateur par l'utilisation de contraintes d'assemblage. La transition entre ces deux espaces correspond à un changement de mode de contrôle dynamique au sein du simulateur physique. L'espace exploratoire est ainsi régi par des lois classiques de

dynamique du contact (voir chapitre 2.2.2), nous proposons d'utiliser des lois de dynamique contrainte dans l'espace d'assemblage.

Dans les mécanismes créés par CAO, la modélisation des opérations d'assemblage peut être réalisée par l'analyse des trajectoires entre des paires de composants. Ces trajectoires se classent en deux familles, selon que l'on considère qu'elle rompe un contact entre les objets ou qu'elle en crée. L'identification de ces contacts permet de définir des classes de mouvements possibles (des mobilités) entre les objets, afin d'en modéliser des liaisons cinématiques. Cette modélisation nécessite généralement une étape d'analyse des liaisons cinématiques formées entre les objets. En effet, pour simuler des opérations d'assemblage, il est indispensable de procéder à une phase d'identification et de modélisation de ces liaisons cinématiques, qui font intervenir de nombreux paramètres tels que les configurations de contact entre les géométries des objets.

Dans le cadre de nos travaux, nous souhaitons extraire ces informations cinématiques, pour les réutiliser comme données d'entrée à notre méthode de guidage contraint. En effet, pour assister l'utilisateur, nous proposons de modéliser les mouvements des objets par des liaisons mécaniques élémentaires. Lors de la phase d'assemblage, les contraintes cinématiques que nous appliquons aux objets CAO modélisent ces liaisons mécaniques, afin de réduire partiellement le nombre de degrés de liberté. À l'image des relations cinématiques qui existent entre les objets d'un assemblage réel, les mécanismes artificiels que nous utilisons autorisent (ou non) des mouvements relatifs entre les objets. Ils consistent en des contraintes bilatérales appliquées entre les objets mobiles et les objets fixes dans lesquels on souhaite réaliser l'assemblage. Seth *et al.* [SVO07] proposent une modélisation similaire pour les assemblages d'objets B-Rep. Dans le cadre de la plateforme SHARP, il étudie particulièrement l'influence de la détection des collisions combinée à des contraintes d'assemblage sur l'insertion haptique d'un pion dans un trou (*peg-in-hole*).

Pour chaque tâche d'assemblage, les contraintes que l'on applique sont définies en fonction :

- des mouvements impliqués dans la tâche ;
- de la liaison mécanique qui modélise ces mouvements ;
- des surfaces en contact qui définissent ce type de liaison.

Étant donné les nombreux paramètres impliqués, nous ne pouvons assurer la création et l'application de contraintes d'assemblage dans tous les cas de figure. Cela dit, nous proposons la modélisation de tâches d'assemblages simples, et notamment les tâches les plus courantes utilisées dans les simulations haptiques. Sans préjuger de tous les scénarii possibles appliqués dans les assemblages interactifs, l'expérience montre que certaines tâches élémentaires, telles que les insertions, sont parmi les plus fréquentes. Notre contribution porte sur ces tâches

élémentaires et ouvre la discussion sur les tâches d'assemblage plus complexes, dont on conjecturera la modélisation par liaisons mécaniques.

Dans le cadre de nos travaux, nous proposons donc la modélisation de tâches d'assemblage simples qui s'appuient sur des liaisons mécaniques de base. Nous considérons ici qu'une liaison n'est applicable qu'à une paire d'objets, afin d'éliminer tout risque d'activation multiple ou d'erreur de montage ; nous préciserons, dans la section 3.3.6 les conditions d'utilisation pouvant déroger à cette hypothèse. Lors de la manipulation d'un objet, on n'active donc que la contrainte qui lui est associée, pour une tâche donnée, en laissant tous les autres guidages désactivés afin d'éviter toute perturbation entre les guides. L'utilisateur voit ses mouvements contraints, via l'interface haptique, par les contraintes appliquées à l'objet auquel il est "attaché".

Pour créer les contraintes et les appliquer dans la simulation dynamique, il est préalablement nécessaire d'utiliser des informations typologiques des assemblages, afin de modéliser les mouvements d'assemblage par les liaisons mécaniques. Cette étape est réalisée en amont de la simulation, dans une phase de pré-processing, où les contraintes cinématiques sont créées et placées dans la scène virtuelle. Cette étape peut être réalisée manuellement en récupérant les caractéristiques géométriques des assemblages sur les modeleurs CAO, quand ces modeleurs le permettent, ou elle peut être réalisée automatiquement en exploitant les informations de contact entre objets. En effet, pour faire correspondre des liaisons mécaniques à des classes de mouvements, il est possible d'identifier les zones d'intérêt par l'analyse des surfaces en contact dans l'assemblage. Dans [IML08] Jacob *et al.* proposent un modèle capable de représenter tous les déplacements autorisés pour chaque couple de surfaces d'une liaison mécanique. Basé sur un modèle de mobilité des contacts élémentaires représentant les rotations et les déplacements dans un mécanisme, ils identifient les différentes interactions entre les composants d'un assemblage. Dans [IML07], ils proposent en particulier un outil, mis en oeuvre dans *Open Cascade*¹, qui permet de capter le type des contacts entre deux composants d'un ensemble mécanique. À partir des surfaces fonctionnelles de contact, il génère des familles de trajectoires auxquelles ils associent des liaisons appui plan, pivot glissant et rotule.

Il est alors possible de qualifier la tâche d'assemblage et de lui associer les liaisons mécaniques, par exemple en la décomposant en tâches simples. En pratique, les liaisons mécaniques qui servent à contraindre les mouvements de l'utilisateur sont modélisables en liaison cinématique appliquée aux objets contrôlés. Une fois que les contraintes et leurs guides associés sont placés dans la simulation, elles restent en attente d'activation, pour le moment où l'utilisateur rentrera dans la phase fonctionnelle d'assemblage. À ce moment, la simulation change auto-

¹www.opencascade.org

matiquement de mode de contrôle haptique, passant d'une physique dynamique autorisant le contact selon 6 DDL, à une physique cinématique restreignant le nombre de DDL (qui dépend de la liaison mécanique active à ce moment.). Dans les deux cas, l'utilisateur se voit contrôlé le nombre de ses mouvements possibles, en gardant à chaque instant le retour d'effort classique inhérent à la simulation : *i.e.* les efforts de contact retournés à l'interface, quelque soit le mode de calcul dynamique (resp. pénalisé) ou cinématique.

Dans les paragraphes suivants, nous présentons chaque liaison normalisée (Norme EN 23952) en fonction des différents types de surfaces fonctionnelles qu'elles impliquent. Nous associons, pour chaque cas, les tâches d'assemblages haptiques qui peuvent être ainsi modélisées ; cette correspondance est primordiale pour définir les contraintes cinématiques à appliquer en fonction des tâches élémentaires à réaliser. Cependant, lors de la modélisation de certaines tâches d'assemblage réelles, nous considérons qu'il n'existe pas de correspondance tâche / liaison élémentaire, si par exemple on se retrouve confronté à des configurations de contacts particulières qui n'appartiennent ni à l'ensemble des contacts élémentaires, ni à l'association de deux (ou plus) contacts élémentaires.

Ainsi, notre but est de définir un modèle capable de représenter, par des liaisons, la plupart des déplacements (rotations, translations, mouvements hélicoïdaux) autorisés pour chaque couple d'objets lorsqu'ils sont mis en contact. On considère ainsi onze types de contacts mécaniques élémentaires qui sont définis par les différents types de surfaces en contact. Pour chaque description, nous précisons l'intérêt et la faisabilité de la modélisation pour les assemblages haptiques :

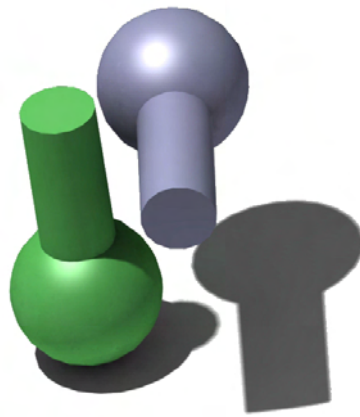


FIG. 3.9 – Vue 3D d'une liaison nulle entre la pièce grise et la pièce verte

La liaison nulle (6 DDL) : Cette liaison correspond à l'absence de contact entre deux objets. Cette liaison comporte 6 degrés de liberté : 3 translations

et 3 rotations. Elle peut être décrite comme l'absence de contact géométrique entre deux objets. Sa considération est faible sur le plan cinématique puisque aucun mouvement ni aucun effort ne lie les objets qui compose cette liaison (voir figure 3.9).

Dans la simulation haptique, cette liaison correspond à la manipulation libre de l'objet dans l'espace, quand les objets ne sont pas en contact. Cette liaison ne correspond pas à une tâche d'assemblage en particulier, mais elle peut être considérée dans le positionnement d'objets dans l'espace. Durant la phase exploratoire de la simulation d'assemblage, on considère que cette liaison est "active" par défaut, mais ne conduit à aucune modélisation cinématique particulière.

La liaison encastrement (0 DDL) : Cette liaison, aussi appelée liaison totale, correspond à une solidarisation complète de deux objets. Cette liaison comporte 0 degré de liberté. Lorsque deux pièces sont encastrées l'une à l'autre, la transmission des efforts est supposée totale entre elles. Sur le plan cinématique elle est sans intérêt puisque les pièces ne disposent pas de mouvement relatif possible. Son identification est toutefois fondamentale dans la modélisation cinématique des mécanismes puisqu'elle permet la définition des classes d'équivalence. En effet, cette liaison ne présente aucune direction particulière et peut être modélisée par combinaison de liaisons ou par l'ajout de butées sur des liaisons existantes (voir figure 3.10 pour la modification d'une glissière en encastrement).

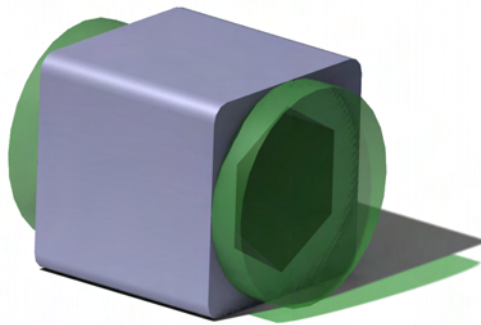


FIG. 3.10 – Vue 3D d'une liaison encastrement entre la pièce grise et la pièce verte

Dans la simulation haptique, cette liaison ne représente pas de tâche d'assemblage en particulier. Les seules liaisons encastrement que l'on peut considérer sont celles qui unissent les objets, s'ils sont formés de sous-assemblages rigides. L'encastrement est ici considéré comme un objectif plutôt qu'un moyen. En effet, lorsqu'un objet est haptiquement assemblé, on peut souhaiter le laisser en place pour s'occuper d'un autre objet, on peut alors créer une liaison encastrement sur

l'objet juste assemblé. Toutefois, nous ne proposons pas de modèle d'assemblage basé sur cette liaison.

La liaison ponctuelle (5 DDL) : Cette liaison décrit un contact entre deux solides qui se réduit à un point. Cette liaison comporte 5 degrés de liberté : 2 translations et 3 rotations. Le contact autorise la transmission d'une force dans la direction normale au plan tangent commun aux deux surfaces en contact. L'axe principal de la liaison est ainsi donné par cette direction et la localisation du point de contact. La liaison ponctuelle peut être considérée comme unilatérale dans la mesure où contact ponctuel interdit le rapprochement des corps mais ne s'oppose pas à leur écartement. Le mode de comportement d'une telle liaison se rapproche du comportement d'un contact haptique.

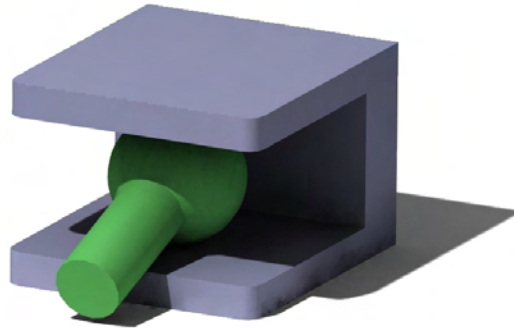


FIG. 3.11 – Vue 3D d'une liaison ponctuelle bilatérale entre la pièce grise et la pièce verte

En règle générale, la modélisation de cette liaison suit un comportement bilatéral qui ne peut s'obtenir que par l'association d'une autre ponctuelle de même normale mais de sens opposé (voir figure 3.11). C'est sous cette forme que nous proposons d'utiliser cette liaison. Pour différencier un contact ponctuel classiquement obtenu par manipulation haptique, il est en effet intéressant de contraindre ce point de manière bilatérale pour interdire, par exemple, le décollement. Nous l'utilisons également en complément d'autres liaisons pour contraindre les mouvements d'un objet, tout en assurant une liberté de mouvement partiel de l'utilisateur.

On se propose d'appliquer une contrainte cinématique ponctuelle entre des points clés des géométries, tels que des sommets d'arêtes, avec les surfaces sur lesquelles elles sont en contact. Nous proposons également, dans une moindre mesure, de l'appliquer sur le point de l'objet qui rentre en premier contact avec un autre objet. Dans les deux cas, on notera l'importance de la méthode de détection des collisions, qui doit être capable d'identifier un point "unique" de contact lors d'une collision entre deux maillages 3D.

Cette liaison est fondamentale pour notre approche VCG parce qu'elle est génératrice de toutes les autres liaisons. En effet, une liaison mécanique peut toujours être décrite par plusieurs liaisons ponctuelles, dès lors qu'une association de points de contact dispose des mêmes particularités géométriques. Nous utilisons cette règle, non pas pour modéliser un mouvement d'assemblage par liaisons ponctuelles, mais pour réaliser des guidages géométriques sur un nombre fini de points de contact des géométries (voir section 3.3.3).

La liaison linéaire rectiligne (4 DDL) : Cette liaison est obtenue par l'association de plus de deux liaisons ponctuelles simultanées. Cette liaison comporte 4 degrés de liberté : 2 translation et 2 rotations. Elle présente un ensemble de points de contact alignés dont les normales sont parallèles. La nature unilatérale de cette liaison fait qu'elle est peu fréquente dans les assemblages mécaniques. Elle y est, en revanche, trouvée en association avec d'autres liaisons. Elle se trouve également sous sa forme bilatérale où le mouvement associé est celui du coulisement d'un pion entre deux plans parallèles (voir figure 3.12).

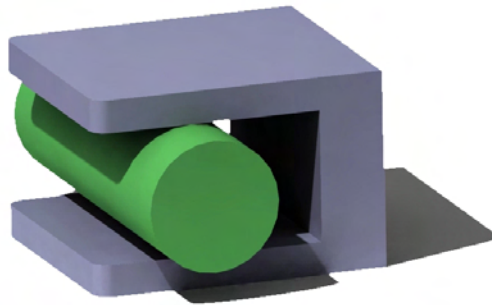


FIG. 3.12 – Vue 3D d'une liaison linéaire rectiligne entre la pièce grise et la pièce verte

D'un point de vue haptique, cette liaison correspond à la configuration de contact entre l'arête vive d'un objet 3D et une surface plane. En général, l'utilisateur a la possibilité de maintenir ce contact via l'interface à retour d'effort, assurant ainsi le maintien de l'unilatéralité de cette liaison. Il ne semble donc pas nécessaire de modéliser une tâche d'assemblage par contrainte cinématique.

La liaison appui plan (3 DDL) : La liaison appui plan décrit un contact entre deux surfaces de sorte qu'elles restent dans un même plan. Cette liaison comporte 3 degrés de liberté : 2 translations et 1 rotation. Cette liaison s'obtient naturellement en plaquant deux surfaces planes l'une sur l'autre de manière unilatérale. Elle se présente également sous forme bilatérale si on considère deux paires de surfaces en contact (voir figure 3.13). Généralement, pour réaliser cette

liaison, il suffit que le contact entre deux solides s'effectue en 3 points au minimum, non alignés (non nécessairement coplanaires) et de normales de direction commune qui donne la direction principale de la liaison.

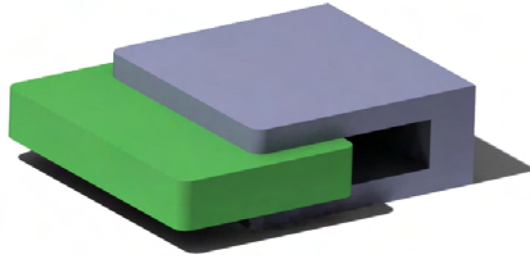


FIG. 3.13 – Vue 3D d'une liaison appui plan bilatérale entre la pièce grise et la pièce verte

Le positionnement d'un objet par appui plan est un mouvement de base de la manipulation haptique. En effet, l'appui plan est modélisable en une contrainte unilatérale, appliquée simultanément sur plusieurs points d'une surface en contact avec une autre. Ce cas est typiquement celui d'un contact conformant, où c'est l'utilisateur qui assure le maintien du contact entre les surfaces. En pratique, il arrive rarement que deux objets aient besoin d'être assemblés selon un appui plan simple. En revanche, cette liaison fait souvent partie de processus de centrage où elle est combinée à d'autres liaisons appui plan, qui servent de butée ou transforment sa nature en une autre liaison (glissière par exemple). C'est pourquoi nous ne modéliserons pas cette liaison avec des contraintes cinématiques, considérant qu'un utilisateur est capable de maintenir deux surfaces en appui tout en les faisant "glisser" l'une contre l'autre, et ce même si ces surfaces sont non visibles ou difficiles d'accès. La liaison appui plan peut, toutefois, intervenir dans un assemblage sous sa forme bilatérale, telle qu'illustrée sur la figure 3.13. Sous cette forme, elle peut modéliser le mouvement de coulissement d'un objet plat entre deux surfaces planes parallèles.

Nous utiliserons les propriétés unilatérales de ce type de liaisons pour introduire le guidage géométrique. Ce guidage consiste en l'ajout des murs virtuels normaux aux surfaces (planes) de l'objet fixe, sur lesquels l'utilisateur vient "se coller" pour suivre une trajectoire prédéfinie. Tout en maintenant le contact plan, il dispose ainsi de suffisamment de liberté de mouvement et des butées pour aligner des objets les uns par rapport aux autres.

La liaison glissière (1 DDL) : La liaison glissière n'autorise qu'un mouvement de translation entre deux objets, selon une direction définie. Cette liaison

comporte 1 degré de liberté : 1 translation. Cette liaison est généralement réalisée à partir de plusieurs liaisons appui plan appliquées simultanément, et qui correspondent souvent aux géométries planes des objets en contact. Cette liaison peut également être réalisée, du point de vue technologique, en bloquant l'axe de rotation d'une liaison pivot-glissant ; c'est l'exemple classique de l'utilisation d'une goupille entre un axe tournant et un alésage.

Suivant les technologies utilisées pour réaliser cette liaison, on peut la considérer comme bilatérale. On peut, par exemple, considérer cette liaison lors de l'insertion d'objets aux formes conjuguées telles que les cannelures ou les queues d'arondes. On peut également garder ses propriétés cinématiques lorsqu'elle est réalisée par deux liaisons pivot-glissant (ou deux liaisons glissières), par exemple pour l'ouverture d'un tiroir.

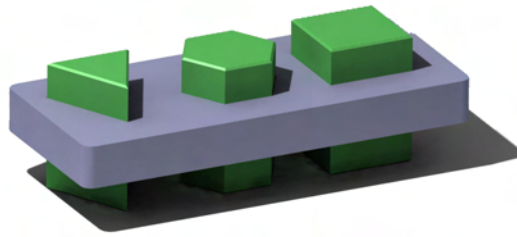


FIG. 3.14 – Vue 3D d'une liaison glissière entre la pièce grise et la pièce verte

Dans les simulations haptiques, nous proposons d'utiliser cette liaison pour les tâches d'insertion d'objets, ainsi que pour la mise en position précise de surfaces. Dans ce dernier cas, elle remplacerait la liaison appui plan bilatérale. Lorsque deux solides sont en liaison glissière, il existe au moins deux droites strictement parallèles communes aux deux solides, dont la direction définit la seule caractéristique géométrique de la liaison. Nous utilisons ces caractéristiques géométriques pour activer des contraintes de type glissière dans les tâches d'insertion d'objets à profils conjugués. Cette notion de profils conjugués, ou complémentaires, est liée aux surfaces en contact lors de l'insertion ; elle s'applique particulièrement bien aux prismes, *i.e.* des volumes extrudés sur profils non circulaires (clavettes, cannelures etc.).

La liaison pivot (1 DDL) : La liaison pivot n'autorise qu'un mouvement de rotation entre deux pièces, selon l'axe de la liaison. Cette liaison comporte 1 degré de liberté : 1 rotation axiale. Elle est souvent obtenue par l'association d'une liaison pivot glissant avec une liaison ponctuelle, ou d'une liaison linéaire annulaire avec une liaison une rotule (montage de roulement à billes). Typiquement, on réalise cette liaison bilatérale en ajoutant un arrêt axial sur un contact

cylindrique sans frottement (deux cônes complémentaires ou deux surfaces de révolution disposant d'une diminution de section).

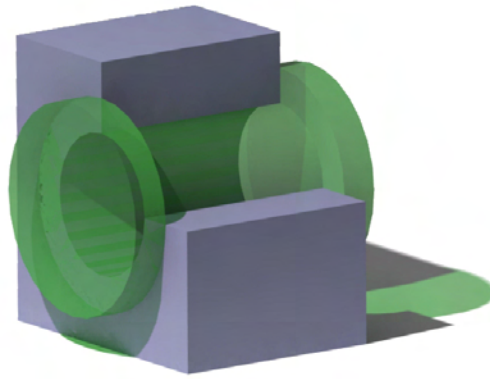


FIG. 3.15 – Vue 3D d'une liaison pivot entre la pièce grise et la pièce verte

Il peut arriver que l'assemblage de deux objets puisse être modélisé par une liaison pivot. Elle interviendra surtout combinée à d'autres liaisons. Cette liaison peut également faire partie d'un processus d'assemblage modélisé par plusieurs contraintes. Il arrive, en effet, que la modélisation d'un mouvement d'assemblage puisse être décomposé en plusieurs étapes séquentielles, chaque étape étant modélisée par une liaison différente (voir section 3.3.6).

La liaison pivot glissant (2 DDL) : La liaison pivot glissant, comme son nom l'indique, combine les caractéristiques des liaisons pivot et glissière, et se comporte globalement comme la combinaison de deux liaisons linéaires annulaires. Cette liaison comporte 2 degrés de liberté : 1 translation et 1 rotation axiale. Elle s'obtient lorsque tous les points de contact entre deux objets appartiennent à un ou plusieurs cylindres coaxiaux, et que toutes les normales de contact intersectent l'axe de ce(s) cylindre(s) : c'est l'axe de la liaison. Cette liaison est très courante dans les mécanismes et est généralement moins contraignante à réaliser qu'une liaison glissière. Si la rotation peut facilement présenter une amplitude infinie (au moins un tour), la translation se limite aux dimensions des pièces.

Dans ce chapitre, nous basons la plupart de nos exemples sur l'insertion d'un pion (*peg*) dans un trou de même diamètre (*hole*), qui est basiquement modélisable en une liaison pivot glissant entre les deux objets. Cette liaison se retrouve souvent dans les assemblages mécaniques (montage de roulement), ainsi que dans les simulations haptiques d'assemblage où elle se prête particulièrement bien aux insertions d'objets à formes conjuguées, dont font partie les prismes à base circulaire (goupilles, axes, pion, etc.).

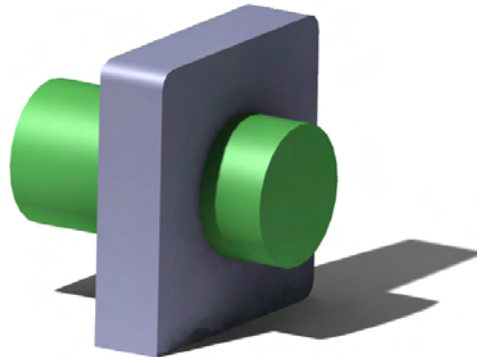


FIG. 3.16 – Vue 3D d'une liaison pivot-glissant entre la pièce grise et la pièce verte

La liaison hélicoïdale (1DDL) : En présence d'un système vis-écrou, on parle de liaison hélicoïdale. Cette liaison comporte 1 degré de liberté : la rotation est simultanée à la translation dans un rapport qu'on appelle le pas de vis, d'hélice ou de filet. De ce fait, il s'agit d'un seul et même degré de liberté. Cette liaison, simple en apparence, fait intervenir des surfaces de contact difficiles à modéliser. La considération de cette liaison est primordiale dans l'étude des dispositifs vis-écrou dont le but est de transformer un mouvement de rotation en mouvement de translation.

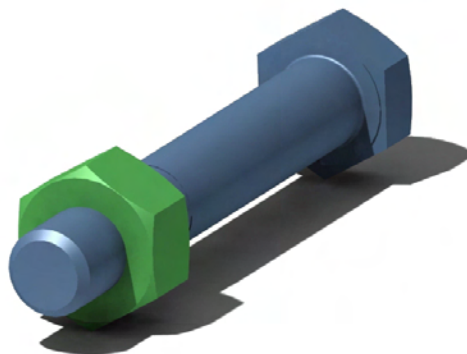


FIG. 3.17 – Vue 3D d'une liaison hélicoïdale entre la pièce grise et la pièce verte

L'étude d'un assemblage par boulons ou vis ne nécessite pas cette modélisation. De même, les maillages représentant ces éléments ne comportent que très rarement les détails de filetage. Notons que ces détails augmentent inutilement le nombre de faces de ces objets standard, et qu'ils ne sont généralement pas traitables par les méthodes de détection classique. Dans la simulation haptique, il n'est pas nécessaire de simuler le mouvement combiné de l'hélicoïde pour les

assemblages de vis ou des éléments standard. On considère le mouvement de vis-sage suffisamment naturel pour ne pas le modéliser avec cette liaison. Toutefois, dans les assemblages où la position du *système vis-écrou* a de l'importance, il est possible et simple de modéliser une liaison hélicoïdale avec un pas de vis symbolique.

La liaison rotule (3 DDL) : La liaison rotule lie complètement deux pièces en translation et nullement en rotation. Cette liaison comporte 3 degrés de liberté : 3 rotations. Elle est l'association de ponctuelles non coplanaires dont les normales s'intersectent en un même point. Ce dernier constitue le centre de cette liaison. Le cas le plus simple est celui de deux sphères mâle et femelle, imbriquées l'une dans l'autre. En pratique, la liaison rotule est souvent associée à une autre liaison rotule pour former une liaison pivot (cas des roulements), ou est utilisée de manière unilatérale pour centrer des pièces entre elles.

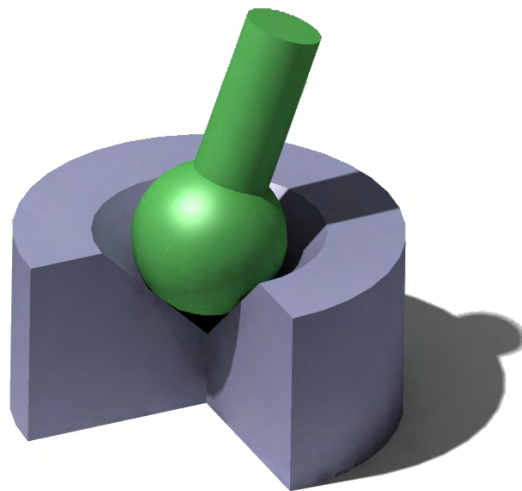


FIG. 3.18 – Vue 3D d'une liaison rotule entre la pièce grise et la pièce verte

La liaison linéaire annulaire (4 DLL) : La liaison linéaire annulaire combine les liaisons rotules et glissière entre deux objets. Cette liaison comporte 4 degrés de liberté : 1 translation et 3 rotations. Elle est obtenue lorsque le contact est réparti suivant un ensemble de points coplanaires et dont les normales de contact sont concourantes. Cet ensemble est un cercle si on dispose une sphère dans un cylindre de même diamètre. Les normales de contact se rencontrent alors au centre de la sphère qui se confond avec le cercle des points de contact. On obtient un équivalent en disposant deux ponctuelles aux normales concourantes, par exemple, une même sphère en contact sur deux plans solidaires et sécants.

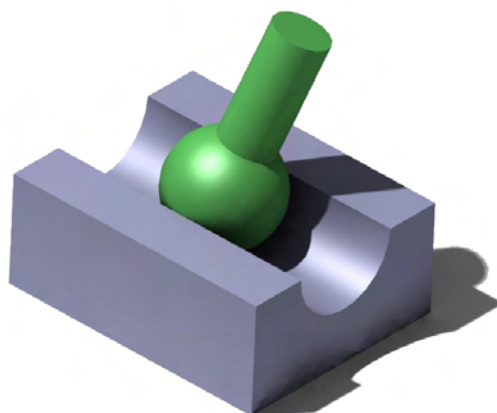


FIG. 3.19 – Vue 3D d'une liaison linéaire annulaire entre la pièce grise et la pièce verte

3.3.2 Guidage géométrique d'objets 3D par entités virtuelles conjuguées

Afin de pouvoir activer les contraintes au sein de la simulation, il est préalablement nécessaire de s'assurer que les objets sont positionnés dans des configurations spatiales compatibles avec l'activation des contraintes. Pour assurer le positionnement relatif des objets avant de les contraindre entre eux, on assure des correspondances géométriques dans l'espace de représentation. Nous proposons ainsi de faire correspondre les attributs géométriques des objets, sous-entendu des objets à assembler, avec des attributs géométriques des contraintes.

Par exemple, avant d'activer une liaison hélicoïdale modélisant le vissage d'une vis, on doit s'assurer au préalable qu'elle est positionnée au-dessus du trou taraudé correspondant, de sorte que leurs axes respectifs soient coaxiaux. Une condition nécessaire et suffisante est l'alignement de l'axe du pion avec celui de la contrainte hélicoïdale. Notons toutefois que cet exemple n'a pas de sens applicatif réel, étant donné qu'il est souvent inutile de simuler le vissage d'une vis.

Cet exemple, qui fait intervenir des conditions de coïncidence d'axes, soulève toutefois le problème de précision des représentations géométriques. Afin de ne pas subir les effets indésirables des maillages "imprécis" des objets (voir section 3.2.1), les positions et orientations des axes de référence doivent correspondre à des données exactes d'assemblage. Plus généralement, les attributs géométriques choisis pour chaque objet doivent être issus de données géométriques du modèleur CAO. En effet, pour déterminer le positionnement des objets dans la scène, on ne considère plus les informations issues des maillages mais des repères-objets que l'on considère fidèlement transférées de la CAO à la RV. Dans le cas où ses objets sont issus de différents modèleurs ou de différents concepteurs on doit alors porter une

attention particulière à la correspondance de ces repères-objets.

Pour aider l'utilisateur à positionner les objets CAO, sous-entendu "objets issus de CAO", nous proposons donc la création et l'application de géométries virtuelles dans l'assemblage, dont le rôle est de contraindre leurs mouvements. Ces *géométries guides*, que nous appelons encore *guides géométriques*, sont utilisées dans le cadre de notre méthode de *guidage géométrique*. Elles ont pour but de limiter les mouvements des objets pilotés, en réduisant localement l'accès à des zones de l'assemblage. Elles consistent ainsi en des extensions complexes des *murs virtuels* tels que ceux proposés par Bettini *et al.* [BML⁺04].

Pour les cas simples d'assemblage, nous considérons deux groupes de guides géométriques. Le premier groupe est constitué des guides liés aux objets "statiques" de l'assemblage. Ces guides, dont la géométrie est discutée chapitre 3.3.3, sont positionnés par rapport aux objets fixes de la simulation, ceux dans lesquels on veut généralement insérer un objet. Le deuxième groupe est constitué des guides liés aux objets pilotés haptiquement. Ces géométries sont fixes par rapport aux repères propres des objets mobiles. Pour les deux groupes, on considère, dans un premier temps, que les objets et leurs guides sont encastrés les uns aux autres, de sorte que tous les efforts appliqués aux guides géométriques sont transférés aux objets et, par extension, aux utilisateurs.

3.3.3 Description des géométries de guidage

Nous proposons la création et l'utilisation de guides virtuels géométriques dans des simulations haptiques d'assemblage d'objets 3D. Dans le cadre de nos travaux, nous modélisons des surfaces impédantes ajoutées à la scène virtuelle.

Ce guidage permet à l'utilisateur de manipuler, positionner et assembler des objets pilotés avec une assistance visuo-haptique. Les guides que nous proposons d'utiliser sont des géométries 3D qui interagissent, en termes de détection de collision, avec des représentations (modélisations) des objets 3D de la simulation, afin de contraindre l'utilisateur à manipuler les objets selon des directions ou trajectoires prédéfinies.

Le guidage géométrique que nous utilisons est réalisé, pour chaque tâche d'assemblage, par une combinaison de géométries 3D ajoutées à la scène, visible ou non de l'utilisateur, qui "forcent" l'utilisateur à positionner l'objet qu'il contrôle dans une position prédéterminée. Généralement proposés par paires, ces guides géométriques sont liés à chaque objet à assembler et leurs positions sont déterminées avant la simulation en fonction des liaisons à réaliser. Nous proposons, pour les tâches simples d'assemblage, de considérer les paires de guides suivantes :

- un guide géométrique est lié en position (et effort) à l'objet piloté par l'utilisateur. Ce guide est donc mobile dans l'environnement ;
- un guide géométrique est lié en position (et effort) à l'objet dans lequel on

souhaite réaliser un assemblage. Ce guide est donc fixe dans l'environnement. L'objet auquel il est attaché est généralement également fixe dans les simulations haptiques classiques.

Ces guides géométriques peuvent être considérés comme des extensions ou, nous le verrons plus loin, des restrictions rigides des objets à assembler, de telle sorte que l'utilisateur manipule "naturellement" ces objets sans être gêné par leurs géométries fictives.

Étant donné que ces guides sont attachés à des objets sur lesquels les lois de contact sont normalement appliquées, on considère les géométries guides comme étant elles-mêmes des objets potentiellement en contact. Les objets pilotés étant encastrés aux guides géométriques "candidats", et les objets fixes encastrés aux guides géométriques "antagonistes", les lois de contact s'appliquent normalement sur ces guides. Ainsi, si le guide géométrique mobile entre en collision avec un autre guide (ou une géométrie exacte), l'effort de réaction qui en résulte est ainsi transmis intégralement à l'objet piloté, et donc transmis à l'utilisateur via l'interface haptique.

Pour réaliser un guidage, ces géométries fictives additionnelles interagissent avec le reste de la scène virtuelle en termes de détection de collision, selon deux combinaisons :

- interaction entre un guide géométrique et la géométrie exacte d'un objet à assembler, que nous nommerons "objet-guide" dans la suite du mémoire ;.
- interaction entre deux (ou plus) guides géométriques, que nous nommerons "guide-guide" dans la suite du mémoire.

Suivant le nombre et les géométries des objets, mais surtout suivant les contraintes liées aux assemblages, nous proposons l'utilisation de l'un ou l'autre type d'interaction. Le premier, tel qu'il est présenté dans la littérature, comme par exemple pour des applications chirurgicales utilisant des géométries frontalières [LIT07], est notamment utilisé pour définir des zones interdites de l'environnement et où l'utilisation de la géométrie (maillée) de l'objet a une importance majeure dans l'assemblage. Le deuxième, qui représente notre contribution pour le positionnement haptique d'objets, est notamment utilisé dans les cas où la perception des géométries est moins importante que la performance ou la compréhension globale de l'assemblage.

La formalisation des guides est ainsi primordiale pour l'assistance à l'assemblage, surtout si on veut que les guides virtuels soient utilisés comme de véritables outils d'assistance. Cependant, il n'existe pas de moyen interactif permettant à l'opérateur de créer ses propres guides virtuels, de les configurer ou de les activer, selon la tâche qu'il désire réaliser. Un autre problème lié à l'utilisation de ces guides se situe au niveau de leur portabilité. En effet, les guides virtuels sont très

spécifiques aux applications pour lesquelles ils ont été créés et sont difficilement réutilisables d'une application sur l'autre.

Dans un premier temps, notre approche consiste à définir une librairie de guides virtuels de base dont l'utilisation est limitée aux différentes tâches que nous avons prédéfinies (insertion d'objets). À chaque guide virtuel est associé des propriétés permettant à l'opérateur d'identifier ces outils et surtout de les réutiliser pour d'autres tâches. La création des guides étant réalisée hors ligne, nous conjecturons leur création, paramétrage et leur cas d'utilisation dans nos futurs travaux.

Nous proposons différentes hypothèses concernant la forme des géométries guides. Leur création dépend de :

- la forme géométrique de l'objet cible et ses surfaces en contact lors de son assemblage ;
- la forme de l'objet piloté, son profil (voir section suivante) et ses surfaces en contact lors de son assemblage ;
- les conditions d'accès à l'objet cible ;
- la liaison qui modélise les mouvements d'assemblage.

Deux types de représentation sont alors envisagées :

1. ces géométries sont des objets 3D, dont les formes sont simultanément liées aux géométries des objets à assembler et au type de liaison à réaliser. De formes simples (plan, cylindre, cône, etc.), elles interagissent directement avec les objets CAO de manière à guider ceux-ci, au contact, vers des positions prédéfinies. Nous utilisons particulièrement les surfaces (volumes creux) de ce type de géométrie en les fixant aux objets fixes de l'assemblage.
2. ces géométries sont des objets 1D ou 2D, qui représentent de manière abstraite les géométries des objets à assembler. De formes élémentaires (segments, points), elles modélisent les géométries "utiles" des objets CAO pour les positionner au contact d'autres guides. Ce sont sur ces modèles, plutôt que sur leurs homologues originaux (les géométries des objets CAO), que s'appliquent les interactions de contact.

La figure 3.20, extension de la figure 3.2, replace la création des guides géométriques dans le processus de traitement des représentations géométriques des objets. Le modèle maillé est toujours traité par le simulateur pour le traitement physique et le traitement graphique. Le simulateur crée, en parallèle, les guides géométriques abstraits et les guides géométriques fixes.

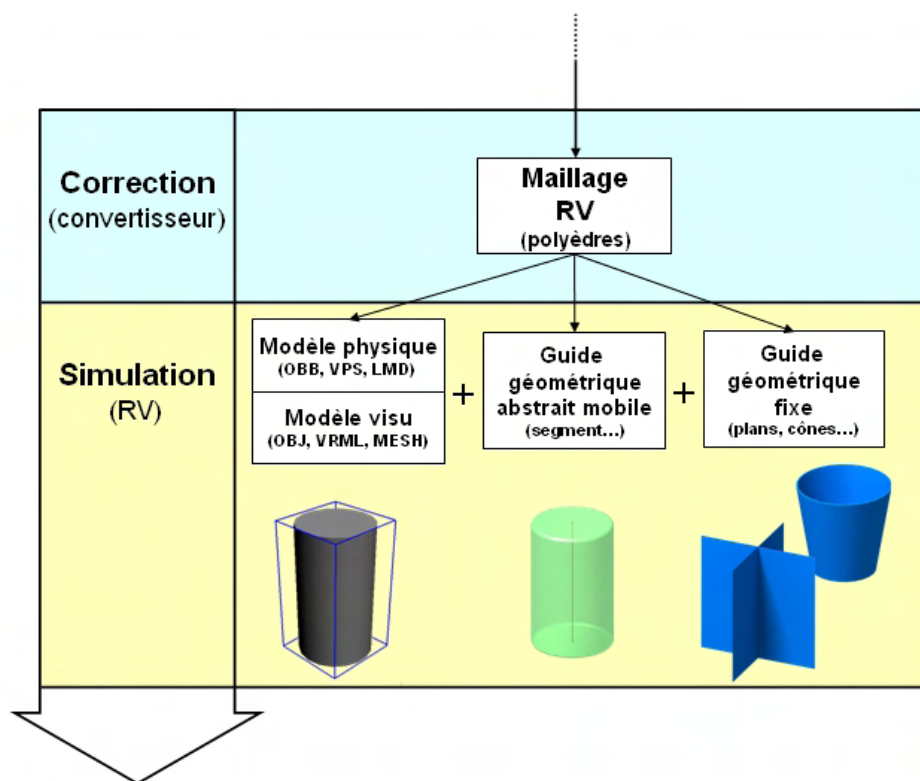


FIG. 3.20 – Création des guides virtuels, au sein du simulateur, à partir des géométries maillées.

3.3.4 Interaction entre les géométries maillées des objets et les guides géométriques 3D

Les guides géométriques sont superposés à la scène virtuelle pour agir comme des *murs virtuels*, au contact desquels les objets voient leurs mouvements restreints. En faisant glisser les objets pilotés contre ces murs, on considère la création locale de contacts unilatéraux ponctuels, surfaciques ou linéaires entre les objets. En créant ainsi des liaisons mécaniques entre les objets (ponctuelle, appui plan ou linéaire), on peut s'assurer de la mise en position complète des géométries.

Tant que les objets sont en phase exploratoire ou en phase d'approche, les contacts entre les objets et/ou entre les guides sont actifs. Tant que les contraintes cinématiques ne sont pas activées (et les collisions désactivées), les guides géométriques sont soumis aux mêmes lois de contact que les autres objets. Au moment où ils entrent en contact avec d'autres objets, le simulateur leur applique des forces de réaction similaires à celles appliquées aux objets en contact.

En disposant de tels *murs virtuels* dans la scène, qui ont une rigidité infinie, on peut ainsi interdire partiellement l'accès à certaines zones de l'espace. Cette

restriction est rendue possible par le contact établi entre l'objet piloté et les surfaces des guides géométriques. On peut alors positionner ces surfaces de sorte à contraindre les trajectoires des objets pilotés. On peut ainsi forcer, ou plutôt contraindre les mouvements de l'utilisateur, du moment qu'il maintient le contact objet-guide (de manière unilatérale) sans interruption. On l'aide ainsi à positionner l'objet dans l'espace, en le laissant "suivre haptiquement" les surfaces des guides. À tout moment, il garde alors la liberté de "décoller" l'objet du guide, par exemple pour essayer une autre stratégie d'approche.

L'exemple le plus simple est celui de la liaison linéaire annulaire (voir figure 3.19) formée entre une sphère et deux plans séquents. Si l'utilisateur contrôle cette sphère, on peut l'aider à la positionner dans l'espace en créant une "rigole" avec les plans. L'utilisateur n'a alors qu'à plaquer la sphère (en double contact) entre les plans et suivre la trajectoire définie par la ligne formée par l'intersection des plans.

L'exemple se complique si on ne considère plus une sphère mais un cylindre. Pour l'aligner contre les plans, et ainsi former une liaison linéaire rectiligne, l'utilisateur dispose de deux types d'approche. L'axe du cylindre arrive soit parallèlement à la ligne d'intersection des plans, soit incliné par rapport à cette même ligne. Dans le premier cas l'utilisateur n'a qu'à plaquer le cylindre sur les plans. Dans le deuxième cas il devra donner au cylindre, une translation jusqu'à entrer en premier contact, puis une rotation qui finalisera le placage du cylindre entre les plans. Cependant l'utilisateur, qui ne décompose pas ses mouvements de la sorte, aura tendance à ne donner qu'un mouvement composé de translations et rotations. Le problème qui nous fait nous attarder sur cet exemple, est que l'utilisateur a souvent tendance à contrôler un objet en translation, et que la rotation requise dans ce mouvement n'est obtenue que s'il relâche la tension de son poignet à l'issue du premier contact cylindre-plan.

Pour les insertions de prismes, une technique de guidage consiste en "l'effet entonnoir" : il s'agit d'ajouter, au dessus des trous, des surfaces qui concentrent les mouvements d'un nombre fini de points de contrôle des objets. Les figures 3.21 et 3.22 illustrent deux exemples de guides fixes à mettre en place pour forcer l'orientation des prismes. Sur la figure 3.21, le pion cylindrique pénètre à l'intérieur du cône surfacique par le dessus. En fonction des caractéristiques géométriques de ce cône (hauteur, rayon de pied et de tête), le pion est plus ou moins facilement positionné. Sur la figure 3.22, le pion en forme de "T" doit entrer à l'intérieur du guide surfacique. Celui-ci forme un entonnoir dont la base est cylindrique et le profil de tête est celui du trou.

Nous proposons des modèles de guidage pour les tâches d'insertion d'objets prismatiques ou pseudo-prismatiques. Par prisme nous considérons tout objet obtenu par extrusion d'un profil de base (pion, clavette, tube, arbre cannelé). Par pseudo-prismatiques nous considérons les objets qui, dans la façon dont ils

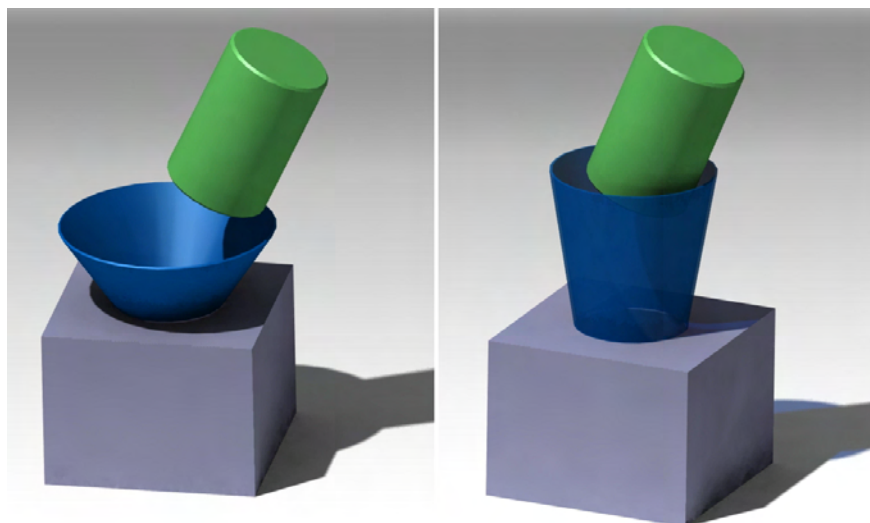


FIG. 3.21 – Guidage géométrique entre un pion cylindrique et une surface conique.

doivent être assemblés, sont modélisables par des prismes dont les profils sont obtenus par projection (roulement à billes). On peut, par exemple, s'assurer du positionnement d'un pion cylindrique au dessus d'un trou en s'assurant que son axe central coïncide à celui du trou. Pour se faire on peut guider la géométrie totale du pion, ou guider directement son axe. Dans le premier cas, on considèrera un guidage "objet-guide" dans l'autre un guidage "guide-guide". On s'assure, de même, du positionnement d'un pion quelconque en s'assurant que les géométries qui composent son profil coïncident à celles du profil cible.

3.3.5 Interaction entre les représentations abstraites des objets et des guides géométriques 3D

Ce phénomène de "suivi de surface", proposé pour les guidages *objets-guides* n'a pas besoin d'être systématiquement réalisé sur les surfaces maillées des objets. En effet, pour les objets disposant de caractéristiques géométriques telles que des symétries, on peut appliquer les mêmes techniques de guidage sur des représentations abstraites des objets.

Reprenons l'exemple de la liaison linéaire annulaire. Pour positionner une sphère dans l'espace on a besoin de connaître ses positions et orientations. Lorsqu'elle crée une liaison linéaire annulaire avec un autre objet (une gouttière par exemple) on ne s'intéresse alors qu'à sa position ; cette position est donnée par la position de son centre, c'est-à-dire celle d'un point dans l'espace. C'est ainsi que l'on propose d'appliquer le guidage géométrique sur le centre d'une sphère plutôt que sur sa surface.

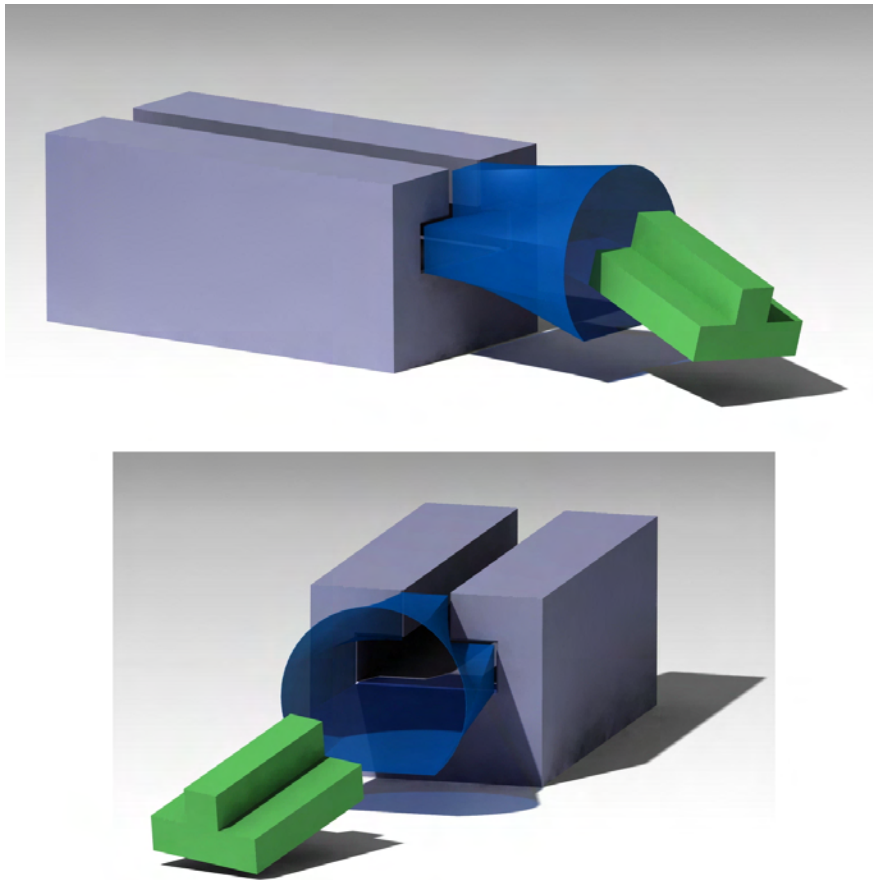


FIG. 3.22 – Guidage géométrique entre un prisme en "T" et une surface complexe.

Par extension, on peut tenter de modéliser des caractéristiques géométriques des objets dans le cadre des liaisons qu'elles forment avec les guides. Plus généralement, on considère :

- le positionnement de deux points dans l'espace pour assurer le positionnement d'un axe ;
- le positionnement de trois points dans l'espace pour assurer un positionnement d'une surface.

La figure 3.23 illustre deux exemples de positionnement de *peg-in-hole*. Sur la figure de gauche, le positionnement d'un cylindre au-dessus de son trou revient à aligner deux segments dans l'espace (soit la coïncidence de deux paires de points dans l'espace). Sur la figure de droite, où le pion est de base carrée, le positionnement revient à aligner trois segments (trois arêtes du cube théorique) dans l'espace.

La figure 3.24 illustre un exemple de guidage géométrique pour le positionne-

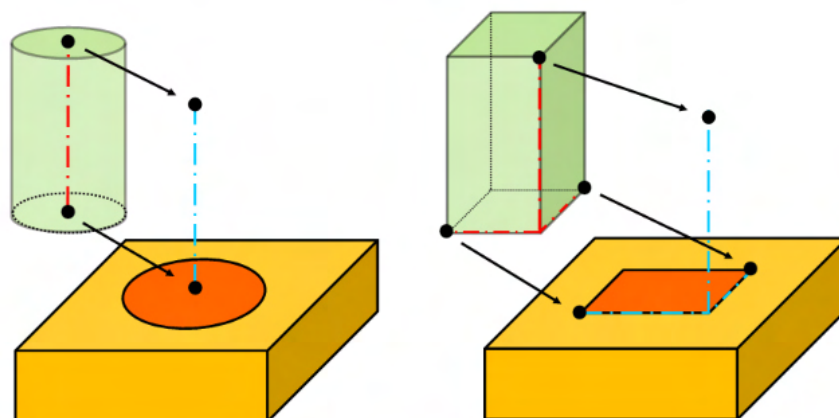


FIG. 3.23 – Positionnement requis pour activer la contrainte mécanique. Sur la figure de gauche, le positionnement d'un pion cylindrique est donné par la coïncidence de leurs axes. À gauche, le positionnement d'un pion prismatique (base carrée) est donnée par la coïncidence de trois arêtes.

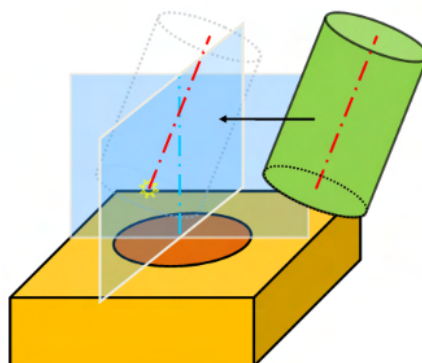


FIG. 3.24 – Exemple de guides géométriques utilisés pour le positionnement d'un pion au dessus d'un trou.

ment d'un pion au dessus d'un trou. Le segment rouge est lié au pion amovible et représente, de manière abstraite, l'axe central du pion. Les plans bleus sont fixes au dessus du trou et représentent les murs virtuels. L'utilisation combinée de ces guides géométriques interdit à l'axe du pion de passer outre les plans. Étant donné l'objectif de l'utilisateur : "approcher le pion vers le trou", la disposition des guides plans l'aide à positionner le centre du pion.

La figure 3.25 illustre l'utilisation du guidage géométrique pour l'exemple *peg-in-hole*. La ligne pointillée verte représente le guide géométrique mobile, fixé sur l'axe du cylindre piloté : une paire de points (un segment). La ligne bleue

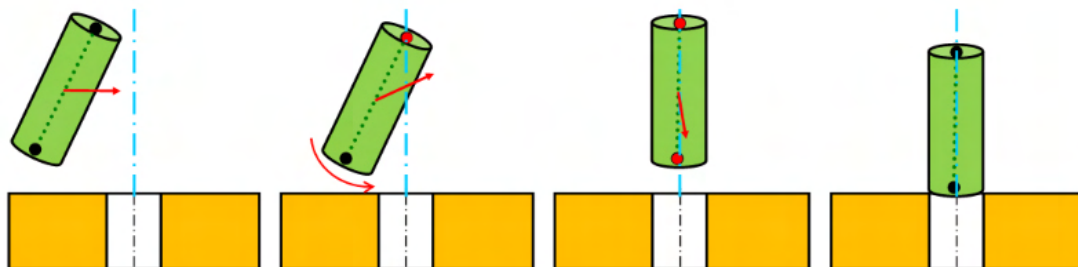


FIG. 3.25 – Illustration du guidage géométrique, vue en 2D, pour l'assemblage *peg-in-hole*.

représente le guide géométrique fixé au-dessus du trou, selon l'axe de la liaison pivot-glissant : un plan. Initialement, (figure de gauche) le pion est piloté par une interface haptique et dirigé vers le trou (on considère que l'on sort de la phase exploratoire). Lorsque les guides entrent en contact ponctuel (figure du milieu) la combinaison de la force de réaction locale et le mouvement de l'utilisateur force l'objet à s'aligner contre les guides géométriques fixes. Une fois que les axes des guides sont concourants, il reste à l'utilisateur à donner un mouvement vertical vers le bas tout en préservant le contact linéaire entre les guides géométriques (figure de droite).

3.3.6 Propositions sur l'utilisation des guides géométriques

Nous proposons, dans cette section, des éléments de réflexion sur une utilisation différente du guidage géométrique.

En premier lieu, pour une unique tâche d'assemblage, nous proposons de sortir du schéma : un objet = un guide géométrique. En effet, il est envisageable de combiner plus de deux guides géométriques pour une même tâche d'assemblage. Toute tâche d'assemblage peut être décomposée en un nombre fini de sous-tâches. Selon que sa complexité interdise sa modélisation par une liaison simple, ou qu'il nécessite un séquençage imposé, un montage peut être modélisé par plusieurs liaisons mécaniques (simples ou non). Activés puis désactivés successivement, les guides géométriques permettent de guider l'objet selon un séquençage prédéterminé. Une telle approche est comparable aux travaux de Kuang *et al.* [KPZ⁺04], qui utilisent plusieurs géométries creuses (tunnels cylindriques) pour forcer un objet à suivre des trajectoires complexes. En ce qui concerne la tâche d'insertion *peg-in-hole* il est possible d'utiliser plus de deux géométries guides. Si l'alignement de deux axes peut intuitivement se réaliser en positionnant un cône plein dans un cône creux, il ne faut pas écarter l'utilisation d'autres combinaisons de géométries pouvant mener au même résultat. Les formes ou les dispositions des

objets pouvant être complexes (objets convexes, inaccessibles etc.), il est envisageable d'utiliser plusieurs formes simples qui, associées entre elles, conduisent au même résultat.

En second lieu, nous avons posé pour hypothèse que les guides liés aux objets fixes de la scène sont eux aussi fixes. Il est toutefois envisageable de les rendre mobiles. En fonction de la position du guide mobile dans l'espace, il est possible de rendre mobile le guide lié à l'objet fixe. Pour le *peg-in-hole*, nous avons proposé un guide fixe constitué de deux plans intersectés. En les remplaçant par des demis plans, leur intersection, qui constitue la géométrie d'intérêt pour notre guidage géométrique, reste une ligne. En fonction de la position du pion, le quartier formé par les deux demis plan peut être mis en rotation de sorte qu'il soit toujours face au pion (voir figure 3.26).

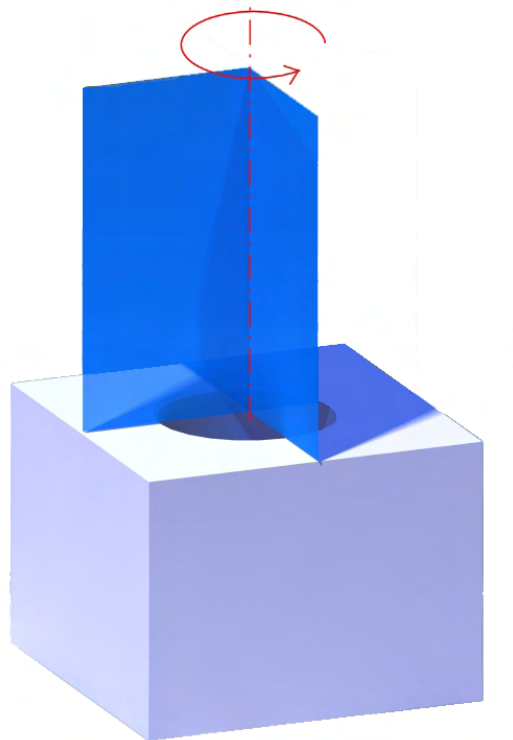


FIG. 3.26 – Guide géométrique rotatif pour l'application *peg-in-hole*.

3.3.7 Changement de mode de contrôle : activation du guidage contraint

Le rôle du guidage géométrique est la mise en position (et orientation) de l'objet piloté dans une configuration où la contrainte mécanique est directement activable. Une fois les objets positionnés, l'application se charge de désactiver les collisions entre les objets concernés et d'activer la contrainte cinématique qui correspond au type d'assemblage souhaité. Dans l'exemple du *peg-in-hole*, la contrainte de pivot glissant est activée lorsque les axes respectifs du pion et de la liaison (celui-ci étant confondu à l'axe théorique du trou) coïncident. La contrainte cinématique, qui a été calculée et placée en amont de la simulation, est alors appliquée entre l'objet piloté et l'objet fixe, sans interruption de la simulation en temps-réel. La transition entre le mode de contrôle dynamique libre et le mode de contrôle dynamique contraint correspond alors au changement de mode de calcul des efforts. Pour passer de l'un à l'autre, il est nécessaire d'assurer la continuité des vitesses des objets entre le moment de la désactivation des collisions et l'activation des contraintes tout en assurant la stabilité et la transparence haptique pour l'utilisateur.

Comme nous l'avons présenté au chapitre précédent, les efforts d'interaction entre objets sont déterminés à partir des informations de collision entre leurs maillages. Lorsqu'ils sont appliqués directement aux objets issus des modeleurs CAO, les algorithmes de détection rendent toutefois difficile, voire impossible, la réalisation des assemblages. Dans notre méthode, l'utilisation de contraintes d'assemblage à pour rôle de simuler l'action réaliste de l'utilisateur, en s'affranchissant des problèmes inhérents à la simulation des contacts. C'est ainsi que lors d'une tâche de montage, nous désactivons complètement les collisions entre les objets avant d'appliquer une contrainte cinématique entre eux. À l'inverse, pour une tâche de démontage nous réactivons les collisions au moment de la désactivation de la contrainte. Cette transition est complètement transparente pour l'utilisateur, si :

- la manipulation contrainte fournit les mêmes degrés de liberté que les mouvements réels attendus par l'utilisateur, et dans ce cas la qualité du guidage est imputable au choix de la contrainte et des guides géométriques ;
- aucune instabilité haptique n'est perceptible au moment de la transition entre les deux modes de contrôle, et dans ce cas la qualité du guidage est imputable à la stabilité algorithmique du changement de mode.

Nous décrivons, ci-après, l'utilisation de notre méthode VCG sur l'application canonique du *peg-in-hole*. Sur la figure 3.27, nous présentons trois étapes de l'insertion du pion dans le trou. La ligne du haut présente de manière schématique, l'utilisation théorique du guidage. La ligne du bas présente les captures d'écran

correspondantes, issues de notre plate-forme expérimentale. Sur les figures de gauche, le pion entre en phase de guidage ; les guides sont activés et affichés, tandis que l'utilisateur déplace le pion vers le trou. Sur les figures du milieu, le pion est positionné au-dessus du trou ; les guides disparaissent et les collisions sont toutes désactivées. À droite, on active la contrainte pivot-glissant entre les objets ; le pion devenu bleu est alors contraint et se déplace, avec 2 DDL, jusqu'au fond du trou.

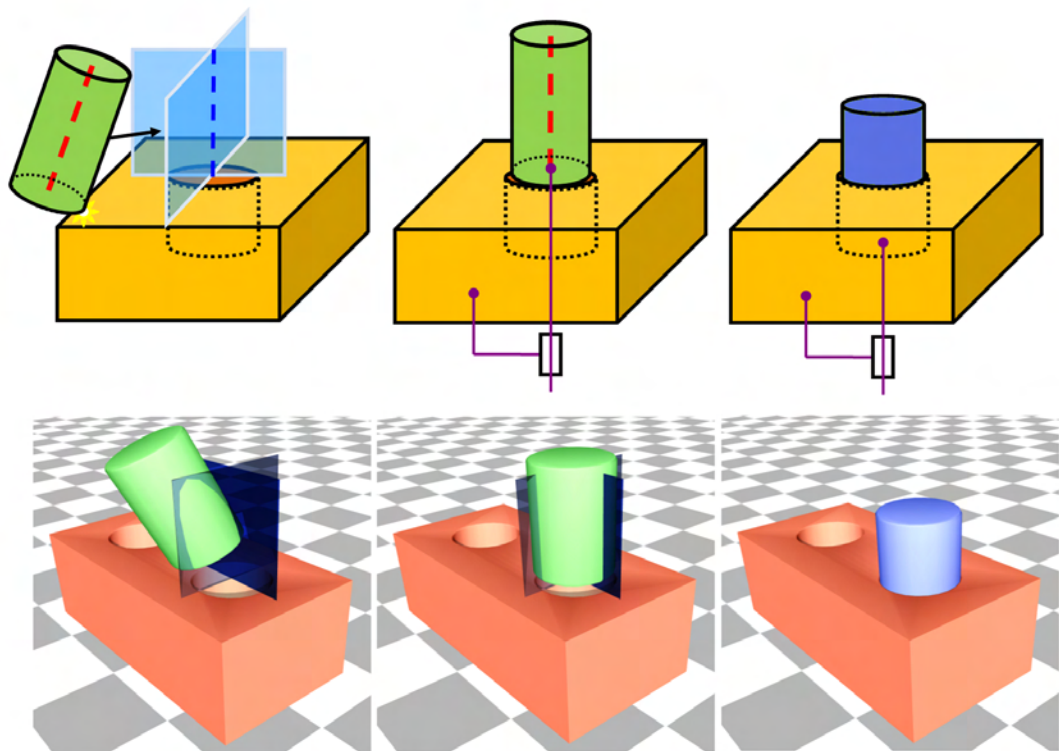


FIG. 3.27 – Les trois étapes du guidage virtuel contraint, vu de manière schématique (ligne du haut) et vu de manière pratique (ligne du bas)

Les figures 3.28 et 3.29 représentent par des graphes à états (grafcets) l'utilisation de la méthode VCG. Le grafcet 3.28 représente le fonctionnement normal du point de vue utilisateur du guidage par VCG. Le grafcet 3.29 représente le fonctionnement normal du point de vue simulateur. Pour les deux grafkets, les paramètres et leurs états associés sont décrits comme suit :

- le paramètre *collision o* représente l'état d'activation des collisions entre les objets CAO ;
- le paramètre *collision g* représente l'état d'activation des collisions entre les guides géométriques ;

- l'état 0 indique que les collisions sont désactivées ;
- l'état 1 indique que les collisions sont activées ;
- le paramètre *pion* représente la position du pion (manipulé) dans l'assemblage ;
- l'état ZE indique la localisation de la zone d'exploration, dans laquelle on ne considère pas d'assemblage ;
- l'état ZF indique la localisation de la zone d'assemblage, dans laquelle on considère que l'assemblage doit être réalisé. L'indication $+/-$ indique la distance plus ou moins proche entre le pion et les plans ;
- l'état PA indique la localisation de la frontière où on active/désactive la contrainte ;
- l'état FT indique la localisation du fond du trou ;
- le paramètre *pion $n+1$* indique que l'utilisateur est attaché à un nouveau pion.

Nous décrivons l'utilisation du guidage virtuel contraint dans le cas où l'utilisateur suit le scénario représenté par le grafcet 3.28. Les différentes étapes de la manipulation guidée sont décrites comme suit :

1. Tant que le pion qui est piloté par l'utilisateur reste suffisamment éloigné du trou (donc de la zone d'assemblage), il interagit "librement" avec l'environnement virtuel. La gestion des contacts est réalisée par des lois de dynamique, ou de quasi-statique, telles que nous les avons décrites au chapitre 2.2.8 ;
2. Une fois que le pion est suffisamment proche de la zone d'assemblage, on considère que les objets sont potentiellement assemblables. On fait alors apparaître les guides géométriques pour signifier la montabilité potentielle des objets, tandis que l'utilisateur continue son mouvement d'avance vers le trou ;
3. L'utilisateur peut alors sortir le pion de la zone d'assemblage, et tout recommencer. Il peut toutefois continuer de déplacer l'objet en direction du trou jusqu'à ce que le guide géométrique mobile, un segment 2D positionné sur l'axe du pion, entre contact avec l'un des guides plans fixés au-dessus du trou. L'utilisateur continue son mouvement d'approche jusqu'à aligner l'axe du cylindre à l'intersection des deux plans fixes. Maintenu plaqué contre l'intersection de ces plans, l'axe du cylindre ne peut suivre qu'une trajectoire rectiligne, et l'utilisateur n'a plus qu'à positionner le pion jusqu'à la base du trou ;

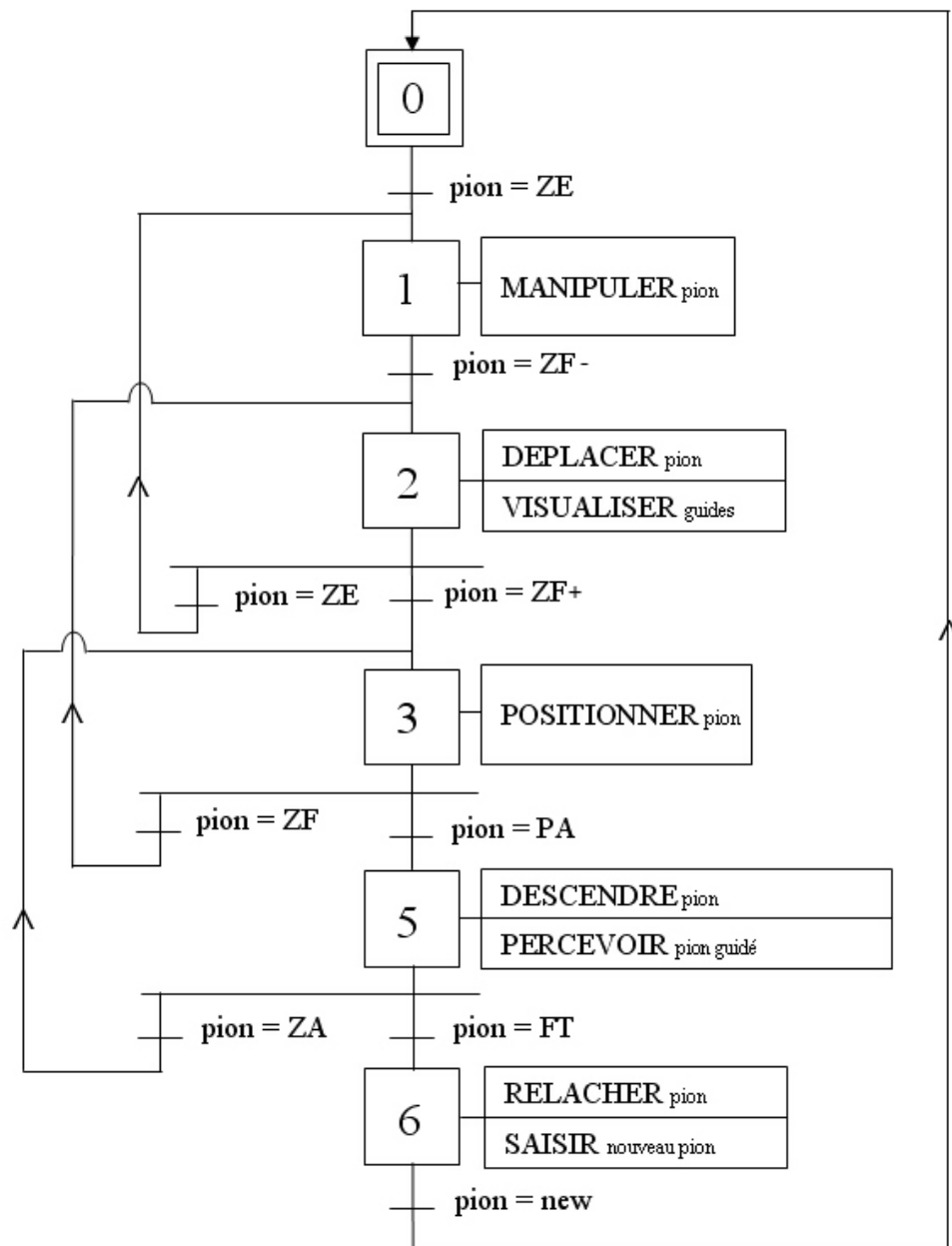


FIG. 3.28 – Grafcet de fonctionnement normal, point de vue utilisateur

4. L'utilisateur note alors le changement de couleur du pion indiquant qu'il est convenablement positionné, c'est-à-dire exactement à la verticale du trou. Il n'a plus qu'à descendre le pion vers le fond du trou pour finir le montage. L'utilisateur peut toujours décider de ressortir le pion et recommencer l'étape de positionnement ;
5. Le pion est convenablement inséré quand il est au fond du trou. Dans ce scénario, l'utilisateur relâche alors le contrôle haptique du pion pour se saisir d'un autre pion et recommencer le montage à un autre endroit de la scène.

Pour ce scénario, le simulateur gère l'activation des collisions, des guides géométriques et des contraintes en fonction de la position du pion dans la scène. Le grafcet 3.29 décrit le guidage VCG du point de vue du simulateur en présentant, de manière algorithmique, les différentes actions à réaliser :

1. Au sein de la zone exploratoire, la gestion des contacts est réalisée par des lois de dynamique (ou de quasi-statique) telles que nous les avons décrites au chapitre 2.2.2. Le simulateur s'assure donc que les collisions entre objets sont bien activées ;
2. Le simulateur teste la position du pion afin de déterminer s'il est en zone exploratoire ou en zone d'assemblage ;
3. Une fois que la distance entre le pion et le trou atteint un certain seuil, paramétré avant la simulation, le simulateur active les collisions entre les guides et les fait apparaître (action non représentée sur le grafcet). Suivant la méthode de guidage choisie (voir section 3.3.3), on peut être amené à activer les collisions objets-guides ;
4. Le simulateur teste la position et l'orientation de l'axe du pion, afin de déterminer s'il est bien positionné au-dessus du trou ;
5. Lorsque le pion est positionné au-dessus du trou, le simulateur désactive les collisions entre les objets, ainsi que les collisions entre les guides. Les guides étant rendus inutiles, le simulateur les fait disparaître. Afin que l'utilisateur rentre dans la phase de guidage contraint, on active la contrainte cinématique entre le pion et le trou. Afin de notifier à l'utilisateur qu'il change de mode de guidage, la couleur du pion est changée (action non représentée sur le grafcet). La contrainte cinématique, modélisant une liaison pivot-glissant entre les objets, est ainsi appliquée. Cette contrainte réduit le nombre de

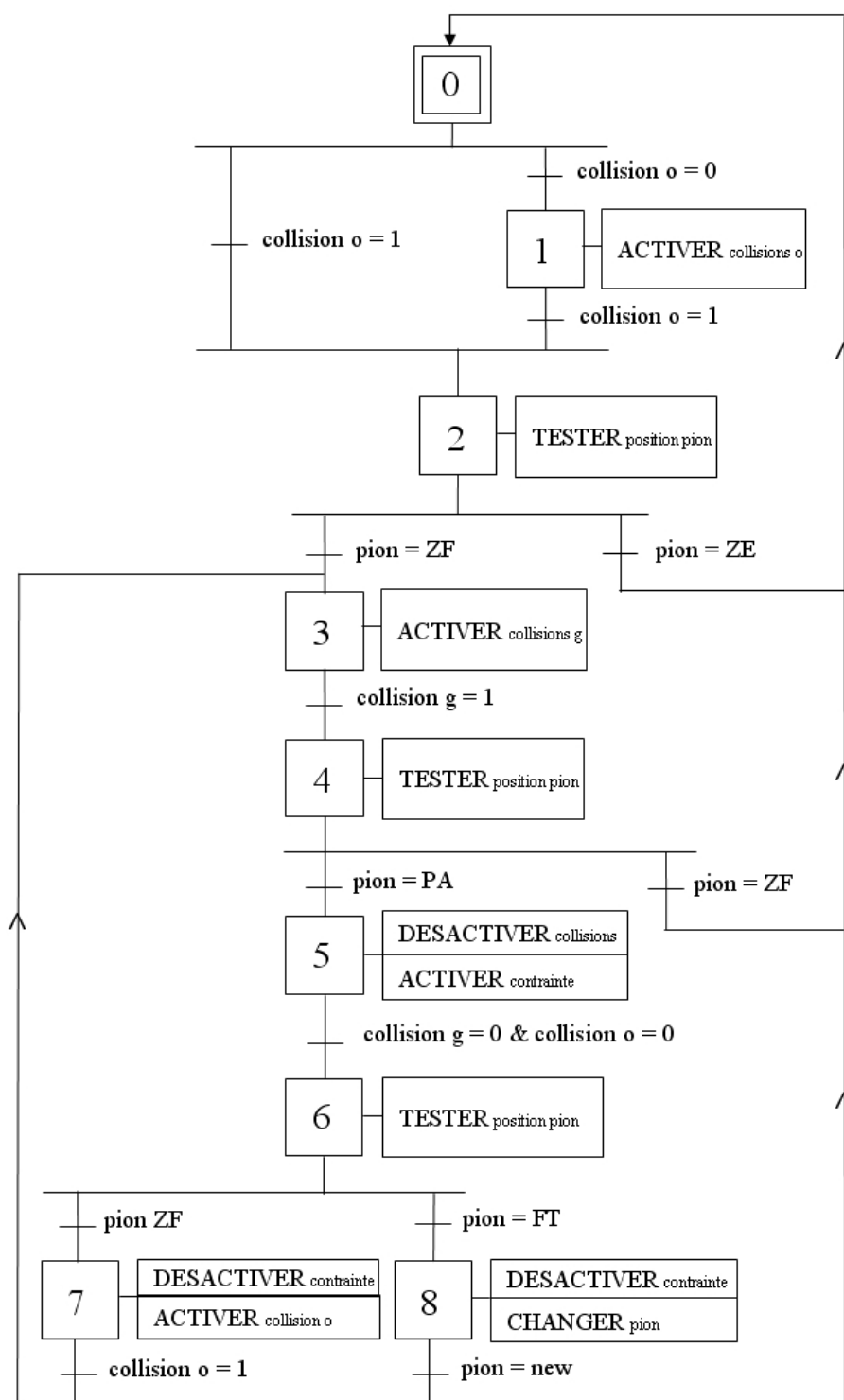


FIG. 3.29 – Grafcet de fonctionnement normal, point de vue simulateur.

DDL du pion à deux, une translation et une rotation, qui correspondent aux mouvements réels autorisés pour une telle tâche d'insertion.

6. Le simulateur teste la position du pion, de sorte que la contrainte reste active tant que le pion n'est pas ressorti ou n'a pas atteint le fond du trou. L'utilisateur n'a ainsi qu'à insérer le pion jusqu'au fond du trou, ce "fond de trou" pouvant être modélisé par une butée virtuelle dans la liaison pivot-glissant ;
7. Si le pion ressort du trou, le simulateur désactive la contrainte et réactive les collisions.
8. Si le pion atteint le fond du trou, la tâche est considérée comme terminée. La contrainte est donc désactivée et le cycle recommence pour un nouveau pion et un nouveau trou. Les collisions n'étant jamais réactivées pour le pion qui vient d'être inséré, celui-ci devient partie intégrante de l'objet fixe et ses guides associés sont complètement désactivés. Une fois découplé du périphérique haptique, on corrige la position de ce pion, si elle est différente de la position d'assemblage théorique.

Durant la dernière étape 8, nous proposons de corriger la position du pion, une fois son assemblage validé. On réalise cet ajustement pour s'assurer de la position "correcte" de l'objet assemblé, en particulier s'il intervient dans des assemblages ultérieurs. Cependant il n'est pas aisé de définir la notion de position correcte. En effet, l'implémentation de notre méthode réalise des mesures de position d'objets 3D pour activer ou désactiver des paramètres fondamentaux de la simulation. Or, si nous connaissons les positions d'assemblage théoriques des objets, nous ne manipulons que des géométries facettisées dont on ne maîtrise pas les erreurs géométriques qu'elles ont avec leurs homologues CAO originaux. Par ailleurs, durant la phase d'approche, le guidage géométrique gère explicitement le positionnement des objets par interactions, au sens du contact, entre les guides géométriques. Or nous avons vu que, en haptique, les algorithmes de détection posent des problèmes de précision spatiale du fait qu'ils utilisent des maillages (parallèles) de moindre précision (voir section 3.2.2). Ainsi, il est nécessaire de corriger, *a posteriori*, les positions finales des objets ; et cela sous réserve de considérer que leurs repères propres sont eux-mêmes *corrects* après les multiples changements de représentation.

3.4 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode d'assemblage d'objets 3D issus de la CAO, dans des simulations de réalité virtuelle. L'utilisa-

tion combinée de *virtual fixtures* et de contraintes mécaniques dans la simulation physique interactive conduit à une réalisation efficace et intuitive d'assemblage simple. Le principal avantage de la méthode VCG tient dans le fait qu'elle permet de travailler avec des géométries directement issues de la CAO et de réaliser des assemblages, en contournant les problèmes inhérents à la détection de collisions.

En ajoutant des guides géométriques aux scènes virtuelles et en les faisant interagir entre eux, nous avons présenté une nouvelle technique de positionnement spatial d'objets virtuels. L'activation en ligne de contraintes cinématiques, autorisée par la coïncidence d'attributs géométriques des objets, assiste l'utilisateur dans la réalisation concrète de tâches d'assemblage. Ces contraintes artificielles, qui agissent sur les mouvements de l'objet par restriction de leur nombre de DDL, sont appliquées pour aider l'utilisateur à suivre des trajectoires préétablies.

Nous avons précisé la transition entre les modes de contrôle haptique entre la phase d'approche et la phase d'assemblage, qui se modélise comme le passage d'une dynamique (des collisions) non contrainte vers une dynamique contrainte. Cette modélisation nous permet de désactiver les collisions entre des paires d'objets 3D et ainsi rendre possible leur assemblage, indépendamment des approximations géométriques engendrées par la représentation géométrique maillée.

En complément, le cas d'étude présenté dans ce chapitre montre la faisabilité de l'utilisation des VCG pour l'assemblage simple d'objets. Dans le chapitre suivant, nous présentons les résultats d'expérience pour l'application *peg-in-hole*, qui montrent l'intérêt de la méthode.

Dans le chapitre suivant, nous présentons une étude anthropocentrée basée sur l'application *peg-in-hole* introduite dans ce chapitre. Nous y évaluons l'utilisabilité de notre méthode de guidage et analysons ses performances temporelles pour la réalisation de tâches d'assemblage.

Chapitre 4

Expériences et démonstrations de la phase exploratoire et de la phase d'assemblage

4.1 Introduction

Nous avons proposé, à la section 1.4, de décomposer la simulation d'assemblage en deux phases distinctes. Chaque phase est définie par la position relative des objets à assembler et des actions à réaliser en fonction de cette position. Deux phases ont ainsi été identifiées : la phase exploratoire qui correspond à la découverte de la scène virtuelle, et la phase fonctionnelle qui correspond à l'assemblage effectif des objets. Pour la partie exploratoire, nous avons proposé, à la section 1.4 de traiter les contacts par dynamique non régulière. De même, la phase fonctionnelle traite les contacts par dynamique contrainte, la transition entre ces deux phases étant assurée par un guidage géométrique.

Dans ce chapitre, nous présentons dans un premier temps Conf&itis, notre plate-forme expérimentale qui couple une simulation physique basée sur la dynamique non régulière avec une interface haptique, pour l'exploration de scènes virtuelles. Cette plate-forme nous permet d'expérimenter la manipulation d'objets simples, et de mesurer les performances algorithmiques de notre méthode de couplage. Dans un deuxième temps, nous présentons des démonstrations de notre méthode VCG pour l'assemblage d'objets issus de la CAO. Nous expérimentons particulièrement l'utilisabilité et les performances temporelles offertes par cette méthode de guidage, sur l'exemple canonique du *peg-in-hole* que nous avons présenté au chapitre précédent. Nous focalisons notre étude sur l'évaluation quantitative et qualitative des méthodes de guidage géométrique et de guidage contraint, unifiées dans sur un même prototype expérimental. Nous concluons alors sur l'utilisation simultanée des deux modes de contrôle dynamique (contraint et non

contraint), discutant de leur avantages par rapport aux méthodes existantes, ainsi que de leurs limites pour leur utilisation sur des assemblages. Nous présentons enfin, section 4.4 des exemples d'utilisation de notre méthode sur des cas d'assemblages CAO réels.

Les méthodes d'exploration et de guidage que nous expérimentons dans ce chapitre sont appliquées sur des données et modèles créés et exportés depuis un modelleur CAO. Pour cette étude, les données 3D et les modèles CAO des objets sont :

- des objets de forme simple (plans, sphères, cylindres etc.), qui nous permettent en section 4.2 de valider le couplage NSCD-haptique et nous servent de guides virtuel en section 4.3 ;
- des objets de forme complexe (surfaces gauches, concaves, convexes) d'origine industrielle, qui nous permettent de présenter nos démonstrations en section 4.4.

4.2 Exploration haptique de scènes virtuelles par dynamique non régulière

Afin d'étudier l'application des méthodes de dynamique non régulière NSCD dans une simulation haptique, nous avons développé un simulateur qui calcule les réactions de collision entre des objets simples de l'environnement. Une présentation détaillée du simulateur haptique est présentée au chapitre 2.3.1. Ce prototype fonctionnel est l'implémentation des approches *time-stepping* de la dynamique non régulière, qui permet de simuler des environnements multi-objets qui sont en interaction. Cette plate-forme logicielle a pour but de représenter des phénomènes physiques réalistes basés sur une application rigoureuse de la mécanique. Elle permet notamment de coupler des interfaces haptiques aux méthodes de traitement des collisions présentées précédemment.

Dans cette section, nous étudions les performances temporelles offertes par Conf&is, en termes de temps de calcul. Pour deux exemples d'application, l'un basé sur l'interaction entre sphères et l'autre sur l'interaction entre parallélépipèdes, nous étudions les performances temporelles des algorithmes de calcul de traitement des collisions, et proposons une évaluation subjective du rendu haptique.

4.2.1 Expériences et résultats

Notre plate-forme expérimentale est couplable aux systèmes à retour d'effort Virtuose (6D-35/45 et Desktop). Pour nos expériences, nous utilisons un Virtuose 6D-35/45. Ce périphérique possède sa propre unité de calcul, qui soulage le processeur de la station de travail. Les contraintes temps-réel sont entièrement prises en charge par un contrôleur associé, qui dialogue directement avec notre application. Le Virtuose combine un retour d'effort selon six axes avec un grand volume de travail, particulièrement adapté aux opérations de manipulation d'objets virtuels à l'échelle 1. Pour l'application que nous avons sélectionné, le *peg-in-hole*, le groupe musculaire sollicité par ce périphérique correspond raisonnablement bien à un cas réel (voir figure 4.6). Aucun *a priori* n'est fait sur la position de la caméra ou de l'utilisateur durant les manipulations.

Nous réalisons nos essais sur un ordinateur portable qui possède un processeur Centrino 2.33 Hz, avec 2 Go de RAM et une carte graphique NVidia Quadro FX2500. L'écran externe de 20" est positionné à approximativement 1 mètre de l'utilisateur, face aux sujets. Notre application n'utilise qu'un seul thread de calcul pour les calculs de détection des collisions et pour le calcul des efforts de réaction. Si nous reconnaissons que c'est là un point faible de notre application, nous obtenons tout de même des performances temps-réels qui autorisent l'interaction. La boucle de rendu haptique ainsi que la boucle de rendu graphique (qui utilise



FIG. 4.1 – Manipulation effectuée sur un bras Virtuose 6D-35/45.

la librairie OpenSource Ogre) tournent sur des threads séparés.

La première expérience met en scène la manipulation haptique d'une sphère. Cette sphère est en interaction avec un nombre déterminé d'autres sphères qui sont en mouvements libres dans un bol (voir figure 4.2). Initialement, les sphères sont disposées au dessus du bol puis lâchées sous l'effet de la gravité. Au bout de quelques secondes, qui dépendent du nombre de sphères et des paramètres de réglage des solveurs, le système se stabilise. Les sphères s'immobilisent sous l'effet des frottements.

L'utilisateur contrôle l'une des sphères dont le diamètre n'a pas d'influence (la sphère verte sur la figure 4.2) via le périphérique haptique. Il contrôle alors les déplacements de la sphère, sans objectif particulier, avec pour seule instruction d'explorer l'ensemble de la scène. Les contacts existant dans la scène sont les contacts sphère-plan, sphère-sphère et sphère-bol. L'utilisateur, une fois qu'il a fait entrer sa propre sphère à l'intérieur du bol, vient "remuer naturellement" les sphères contenues dans le bol. Interagissant ainsi avec un milieu pseudo-granulaire, l'utilisateur ressent les différentes réactions de contact sphère-sphère, les frottements dus aux contacts avec le bol ainsi que les impacts des autres sphères si celles-ci atteignent des vitesses suffisamment grandes.

La deuxième expérience met en scène le contrôle haptique d'une sphère, cette fois en interaction avec un mur de briques. La sphère entre en collision, sans

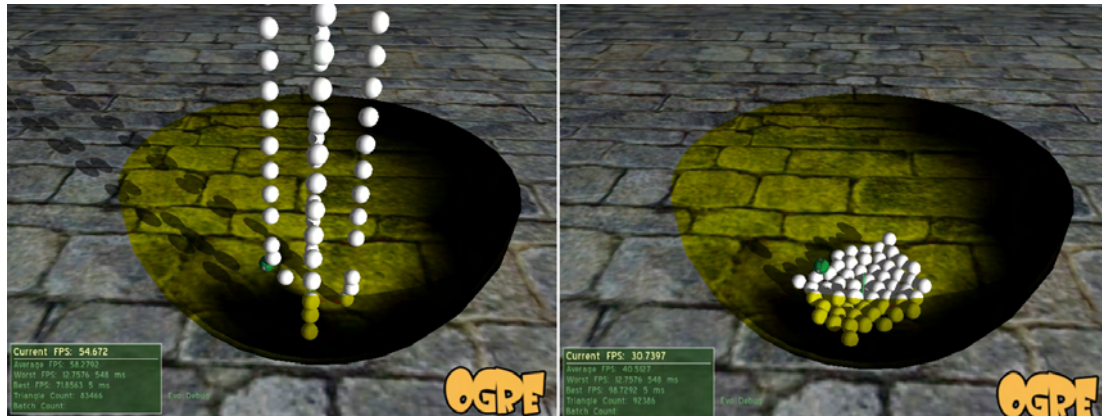


FIG. 4.2 – Interaction entre la sphère (verte) contrôlée par l'utilisateur et 50 autres sphères soumises à la gravité.

aucune interpénétration, avec des parallélépipèdes disposés dans l'environnement virtuel de sorte à faire tomber les briques. La figure 4.2 illustre cette simulation. Initialement les briques sont disposées de sorte à former un mur fixe sur le sol. Lorsqu'aucune interférence n'est appliquée au mur, celui-ci reste complètement stable si sa hauteur ne dépasse pas un certain seuil. L'utilisateur heurte le mur avec la sphère de sorte que l'arrangement stable des briques soit perturbé, et le mur "détruit". En fonction du coefficient de frottement entre les briques et les masses de celles-ci, l'utilisateur a alors plus ou moins de difficulté, au sens haptique, à déplacer les briques pour les faire tomber. Étant donné que notre simulateur ne dispose pas d'un moteur de détection des collisions très performant, le nombre des contacts entre briques est volontairement saturé de sorte à ne pas faire exploser les temps de calculs alloués à la détection. Pour pouvoir respecter la contrainte de temps-interactif et ne pas perturber les retours d'effort à cause de l'insuffisance de la fréquence de calcul du simulateur, le nombre et la dimension des blocs est restreint. Nos essais nous ont permis de monter un mur de 25 pavés (maximum) avant d'atteindre la limite de stabilité de l'interface haptique. Au-delà la simulation est suffisamment lente pour perturber les efforts ressentis.

La figure 4.4 présente les différentes moyennes des performances temporelles de notre simulateur. Elles nous permettent d'étudier l'influence du nombre maximum de contacts simultanés, corrélé au nombre de sphères dans cette application. Nous avons mesuré la fréquence de calcul du simulateur dynamique, en faisant varier le nombre de sphères. Le nombre de pas de calcul par seconde, indépendant du temps de rendu graphique (ici assez sommaire), décroît ainsi significativement en fonction du nombre croissant de contacts. Pour un nombre compris entre 0 et 10 contacts simultanés, les fréquences sont suffisamment élevées pour produire un couplage haptique stable et des efforts cohérents avec la scène.

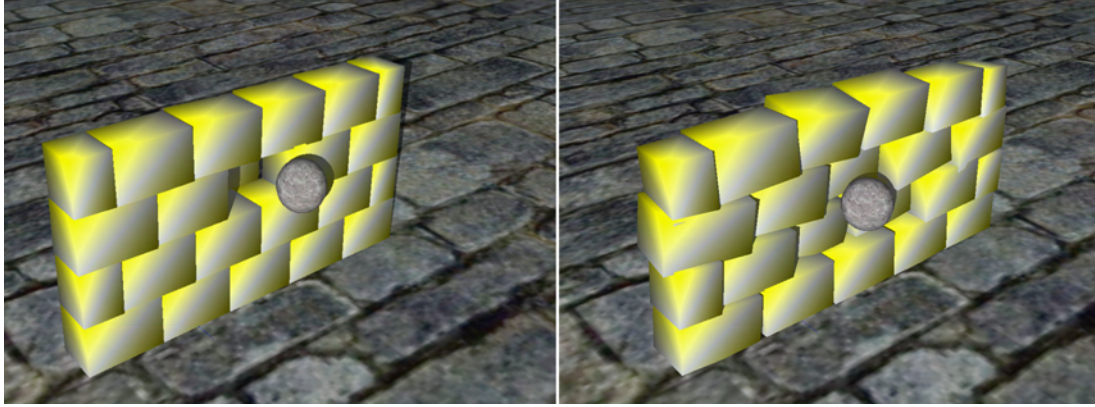


FIG. 4.3 – Interaction entre la sphère (verte) contrôlée par l'utilisateur un mur de 22 briques (jaunes).

Nombre de sphères	Nombre max de contacts	Fréquence de calcul (Hz)
1	1	440
10	25	320
20	44	120
50	160	60
100	250	30
150	350	10
200	430	5

FIG. 4.4 – Estimation des performances temporelles de Confetis, en fonction du nombre de sphères et du nombre moyen de contacts simultanés

L'étude subjective du rendu haptique conforte cette explication. En effet, le rendu haptique est évalué subjectivement par un utilisateur expert en rendu haptique. Cette évaluation porte sur :

- la fluidité (transparence) de la manipulation en l'absence de contact ; cette évaluation donne une référence sur la fréquence de rafraîchissement des efforts de la simulation ;
- la stabilité de l'interface en présence d'un contact simple, c'est-à-dire un contact unilatéral créé à l'interface de l'objet piloté et d'une surface fixe de l'environnement, ou d'un contact ponctuel dans le temps. Notons que ce type de contact est très fréquent dans la simulation sphère-sphère ;
- la précision des positions de contact et des comportements dynamiques de l'objet piloté.

En théorie la construction des scènes des expériences 1 et 2 ne limite pas le nombre d'objets de la scène. Mais pour maintenir une fréquence suffisamment

élevée et ainsi autoriser une interaction haptique stable et pertinente, il est nécessaire d'assurer un maximum de 200 points de contact simultanés. Ce nombre est raisonnablement acceptable si on le positionne dans le cadre de simulations d'assemblage simples où le nombre des contacts, entre surfaces planes notamment, est maîtrisé par l'algorithme de détection des collisions.

Toutefois ces deux expériences nous montrent d'une part la faisabilité du couplage haptique, en vitesse, de l'approche *time-stepping* de la dynamique non régulière. Si les cas que nous présentons restent simples, ils mettent en lumière l'importance des paramètres de réglage et du nombre admissible de contacts à traiter simultanément, pour maintenir des fréquences de calcul compatibles avec les simulations haptiques.

Rendu subjectif :

- plus de 400 Hz : la manipulation est très fluide (transparente), la simulation est stable et les efforts ressentis sont réalistes ;
- de 300 à 400 Hz : la manipulation est fluide, la simulation est assez stable et les impacts sont faiblement ressentis ;
- de 120 à 300 Hz : le rendu haptique est à la limite de l'acceptabilité, la stabilité est incertaine et la transparence est assez mauvaise ;
- de 60 à 120 Hz : le rendu haptique est saccadé. On ressent des vibrations qui ne correspondent pas à des efforts compréhensibles, et des instabilités ;
- de 30 à 60 Hz : limite de perte de contrôle, instabilités et ralentissement de la simulation ;
- de 10 à 30 Hz : perte totale de contrôle haptique, simulation (très) lente ;
- de 0 à 10 Hz : pas de couplage haptique.

4.2.2 Conclusion sur l'exploration des scènes haptiques par dynamique non régulière

Dans cette section, nous avons rappelé les éléments qui composent notre simulateur interactif en soulignant les problèmes spécifiques du couplage simulation-interface. Nous avons alors présenté deux expériences sur lesquelles nous avons mesuré les performances temporelles du simulateur et comparé les résultats avec l'évaluation qualitative du rendu haptique. Elles nous ont permis de conclure sur les plages de fréquence de calcul de notre simulateur, pour procéder à la manipulation haptique normale des objets. Nous avons mis en évidence l'influence du nombre de contacts pouvant être traités simultanément en maintenant la contrainte temporelle de temps-interactif. Ces applications, en plus de montrer la faisabilité du couplage entre une interface haptique et les algorithmes de dynamique non régulière, nous ont permis d'effectuer nos premiers essais sur l'exploration de scènes virtuelles dynamiques. En effet, même si les objets que nous

manipulons sont fondamentalement simples, il est possible de les arranger pour recréer les conditions d'exploration d'une scène de type CAO.

Nous sommes conscients que la plate-forme Conf&tis est purement expérimentale et dispose de nombreuses limites d'utilisation. Celles-ci sont particulièrement liées aux géométries des objets utilisables, et à la stabilité des efforts retournés à l'interface haptique. Nous envisageons dans la suite de nos travaux, d'implémenter un moteur de détection des collisions qui nous permettra d'effectuer des calculs d'effort sur des maillages complexes, plutôt que sur des modèles géométriques analytiques. Par ailleurs des études plus poussées sur les réglages des différents paramètres des solveurs et des lois de contact nous permettront de mieux maîtriser les efforts et vitesses produits par les algorithmes de *time-stepping*. La figure 4.5 illustre, pour conclure, une simulation de dominos où un utilisateur donne la première impulsion par l'interface haptique et est capable d'interagir avec chacun des dominos pour en modifier le comportement dynamique. Cette démonstration est réalisée en environnement "échelle 1" dans le centre de réalité virtuelle de l'IRISA de Rennes.

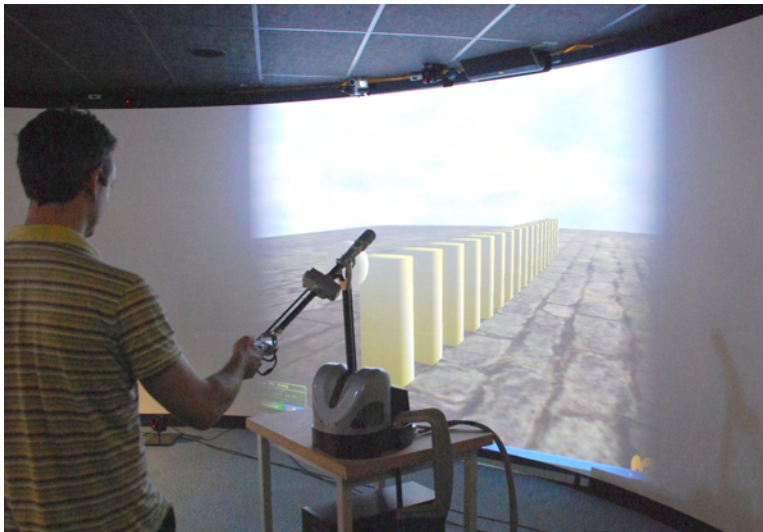


FIG. 4.5 – Interactions, en salle immersive, entre un utilisateur et une ligne de dominos.

4.3 Assemblages haptiques d'objets CAO par guidage virtuel contraint

Nous avons présenté, au chapitre précédent, notre méthode de guidage virtuel contraint pour l'assemblage d'objets CAO en environnement RV. Cette méthode consiste à appliquer localement une contrainte mécanique entre les objets à assembler, tout en désactivant la détection de collisions. Cette contrainte est pré-calculée en début de simulation en fonction des configurations d'assemblage issues de la CAO. Pour parvenir à transiter entre la phase exploratoire et la phase d'assemblage, nous avons également présenté une méthode basée sur des guides géométriques, pour assurer la transition entre les deux différents modes de dynamique. Des géométries virtuelles sont ainsi ajoutées à la scène pour guider les objets dans des positions préétablies.

Dans cette section, nous proposons une étude complète sur les performances et sur l'utilisabilité de notre méthode VCG, appliquée à l'exemple *peg-in-hole*. L'évaluation a essentiellement pour but de vérifier la pertinence du guidage géométrique *guide-guide* tel que nous l'avons proposé à la section 3.3.5, pour l'activation de contraintes cinématiques dans l'assemblage. Nous reconnaissons que l'application *peg-in-hole* ne couvre pas l'ensemble des problèmes rencontrés dans les assemblages interactifs, mais nous sert de base de réflexion sur l'utilisation du guidage virtuel contraint sur des cas industriels réels.

Dans un premier temps, section 4.3.1, nous explicitons les objectifs de notre expérimentation et décrivons le cadre expérimental de notre étude. Puis nous décrivons le protocole expérimental ainsi que les mesures réalisées sur notre application. Nous présentons alors nos résultats expérimentaux et leurs analyses concernant l'utilisabilité du guidage géométrique. Puis nous présentons, en section 4.4, les démonstrations que nous avons réalisées, entre autres, dans le cadre du projet de recherche Part@ge. Nous concluons, au chapitre 4.5 sur l'approche VCG appliquée au cas simple du *peg-in-hole* et discutons de son application sur les cas d'assemblages complexes.

La méthode VCG se décompose en deux parties distinctes qui peuvent être évaluées : le guidage géométrique qui permet d'approcher des objets, et le guidage contraint qui permet de finaliser l'assemblage.

4.3.1 Objectif des expérimentations

Le but de ces expériences est de tester différents types de guidages géométriques pour disposer d'éléments de réflexion sur l'utilisabilité de notre méthode. L'utilisabilité est une notion proche de l'ergonomie qui est définie par la norme ISO 9241 comme "le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et

satisfaction, dans un contexte d'utilisation spécifié". Pour notre expérience, l'utilisateur réalise des tâches de montage / démontage enchaînées, sur lesquelles :

- on vérifie que l'utilisateur peut réaliser les tâches avec notre méthode de guidage (efficacité) ;
- on évalue l'apport de notre méthode sur la performance de l'utilisateur (efficience) ;
- on évalue subjectivement le confort de l'utilisateur lors de la manipulation guidée (satisfaction).

Les expériences que nous réalisons ont ainsi pour objectif d'évaluer les performances du guidage géométrique dans le cadre de la méthode VCG. Cette évaluation porte précisément sur les conditions d'utilisation du guidage *guide-guide* (voir section 3.3.5) pour les tâches d'insertion. Nous souhaitons, d'une part, étudier les gains en performance offerts par notre méthode de guidage. Nous souhaitons, d'autre part, déterminer l'influence des géométries des guides et de leur visualisation dans la scène d'assemblage. Nous souhaitons, enfin, exploiter nos observations sur les approches spatiales des objets, pour optimiser expérimentalement la définition des guides géométriques.

Notre expérience est structurée en deux volets principaux :

- déterminer les performances temporelles d'un utilisateur pour un enchaînement de tâches d'insertion, guidées et non guidées ;
- déterminer l'utilisabilité de notre méthode de guidage.

4.3.2 Plate-forme technologique

Pour nos expériences, nous utilisons un périphérique haptique Haption Virtuouse 6D35-45. Ce périphérique possède sa propre unité de calcul, qui soulage le processeur de la station de travail. Les contraintes temps-réel sont entièrement prises en charge par un contrôleur associé, qui dialogue directement avec notre application. Le Virtuouse combine un retour d'effort selon six axes avec un grand volume de travail, particulièrement adapté aux opérations de manipulation d'objets virtuels à l'échelle 1. Pour l'application que nous avons sélectionnée, le *peg-in-hole*, le groupe musculaire sollicité par ce périphérique correspond raisonnablement bien à un cas réel (voir figure 4.6).

Nous réalisons nos essais sur un ordinateur portable qui possède un processeur Centrino 2.33 Hz, avec 2 Go de RAM et une carte graphique NVidia Quadro FX2500. L'écran 17" du PC est positionné à approximativement 1 mètre de l'utilisateur, légèrement en contrebas mais ajusté à la vue du sujet.

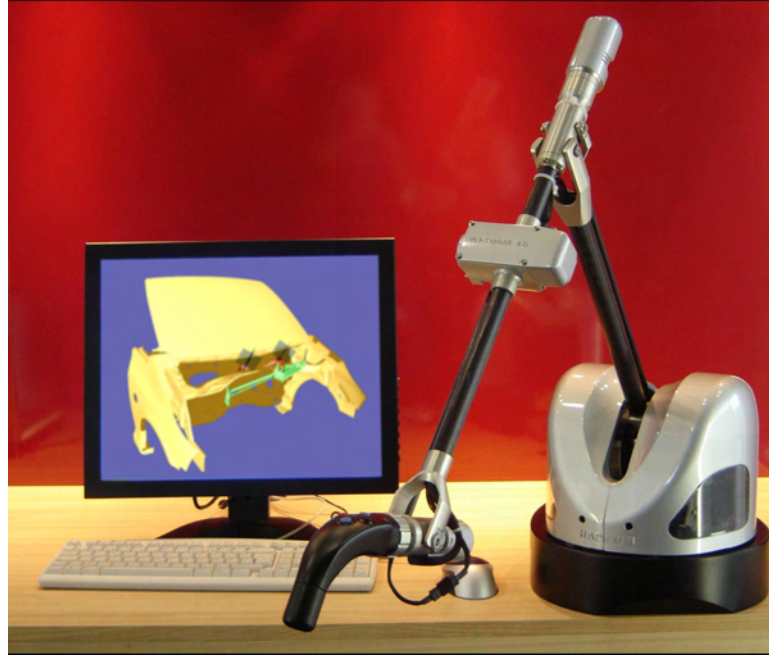


FIG. 4.6 – Assemblage d'objets CAO avec un périphérique haptique Virtuoso.

Pour notre expérimentation, nous utilisons un moteur physique baptisé XDE, pour eXtended Dynamic Engine, développé par le CEA-List. Ce moteur vise à simuler en temps-interactif les évolutions d'un environnement virtuel et permet de :

- détecter les collisions (éviter l'interpénétration de deux solides) et gérer les contacts ;
- calculer les mouvements d'un objet virtuel soumis à des forces externes.

XDE gère les comportements des solides rigides complexes, des chaînes articulées (typiquement un robot), des objets déformables comme les câbles ou les poutres, et des humains virtuels. Profitant de nombreuses années d'expérience sur l'interfaçage haptique, nous avons choisi ce moteur car il propose nativement le couplage des périphériques à retour d'effort Haption. Outre le fait qu'il utilise des algorithmes de couplage haptique stables, il propose la gestion des contacts entre objets par approche quasi statique et dynamique non régulière. Proposant une implémentation des méthodes *time-stepping* pour les objets complexes, son utilisation s'inscrit pleinement dans le cadre de nos travaux. Pour nos expériences, nous utilisons ainsi la méthode LMD++ [MPT07], dont les algorithmes interdisent les interpénétrations entre objets.

4.3.3 Expérience préliminaire sur le guidage contraint

Le guidage contraint consiste en l'application d'une contrainte cinématique entre deux objets sur lesquels on a localement désactivé les collisions. Pour les tâches simples telles que les insertions, cette contrainte peut être activée dès que les positions des objets respectent des critères géométriques associés aux liaisons simulées (alignement d'axes ou de centre etc. voir section 3.3.5). Une fois activée, la contrainte assiste la finition de l'assemblage.

Pour nos expériences, nous avons arbitrairement choisi d'activer une liaison pivot glissant au moment où le pion peut entrer dans le trou, c'est-à-dire lorsque son axe coïncide avec l'axe du trou. Notre expérience consiste à évaluer l'utilisation de la méthode de guidage pour :

- le positionnement du pion (*i.e.* de son axe) au-dessus du trou ;
- l'activation de la contrainte qui suppose le changement de mode dynamique ;
- l'insertion du pion, en mode contraint, jusqu'au fond du trou.

Dans le cadre spécifique de l'insertion *peg-in-hole*, nous posons pour hypothèse que les degrés de liberté d'un pion, lorsqu'il est en cours d'insertion, sont effectivement ceux d'une liaison pivot glissant, et que si nous ignorons les frottements les performances temporelles sont similaires. Cette hypothèse nous permet de restreindre l'étude du guidage aux phases de positionnement et de changement de mode dynamique. L'expérience préliminaire que nous proposons ici permet de valider cette hypothèse. En effet, nous souhaitons montrer que la performance de la tâche est liée à l'approche du pion avant qu'il ne soit dans le trou, et non à l'insertion finale à proprement dit.

Pour montrer cela, nous comparons deux insertions *peg-in-hole*. Dans la première, nous gardons le diamètre du pion tel qu'il est exporté de la CAO, et nous y appliquons une contrainte VCG de type pivot glissant (collisions désactivées). Dans la deuxième, nous avons modifié la géométrie du pion, en fonction de la précision maximale du moteur de collisions (voir section 3.2.2), de sorte qu'il puisse s'insérer dans le trou sans guidage. Les deux modélisations sont représentées sur la figure 3.7. Cette figure illustre les deux cas d'insertion, sans guidage à droite et avec VCG à gauche, lorsque les pions sont au-dessus de leur trou (position de départ). La différence de diamètre du pion, entre les deux cas, est difficilement observable à l'oeil nu. La tâche consiste en :

- l'insertion (descendante) du pion dans le trou, la profondeur du trou étant égale à la hauteur du pion ;
- le ressenti de la butée de fond de trou, qui est soit due à la collision des bases des prismes, soit due à la butée introduite comme paramètre de la contrainte pivot-glissant ;
- l'extraction (ascendante) du pion jusqu'à ce qu'il soit complètement sorti.

Sur trente essais, réalisés par un utilisateur expérimenté en manipulation haptique, le temps moyen pour réaliser la tâche est de 1.33 s pour l'insertion sans guidage, et de 1.25 s pour l'insertion avec le guidage contraint. Cette différence, de 7 %, peut être considérée comme suffisamment faible pour être négligée. Le rendu haptique est évalué subjectivement par l'expérimentateur. D'après son expertise, le ressenti haptique est similaire dans les deux expériences ; la différence entre les deux moyennes temporelles est ainsi imputable aux calculs des collisions effectués sur le cas non contraint.

Nous pouvons ainsi considérer que, pour la tâche *peg-in-hole*, l'assemblage avec guidage contraint offre des performances identiques aux assemblages non contraints. Dans la suite de nos expériences, nous proposons donc de concentrer nos analyses sur le guidage géométrique. L'évaluation du guidage géométrique est pour nous fondamentale, puisqu'elle détermine complètement l'approche des objets avant leur liaison, et puisqu'elle gère la transition entre le mode de dynamique non contrainte et de dynamique contrainte.

4.3.4 Choix de l'expérimentation

La tâche que nous retenons pour notre expérience est l'insertion d'un pion dans un cylindre suivant le modèle *peg-in-hole* que nous avons déjà eu l'occasion de présenter. Nous utilisons cette application canonique car elle est relativement simple à mettre en place et permet d'obtenir rapidement des résultats. Comme nous l'avons vu au chapitre 3.3.1, on peut modéliser cette tâche d'assemblage par une liaison pivot glissant si on considère des formes cylindriques ou une liaison glissière pour les autres types de prismes (droits ou non droits). Elle est fréquemment rencontrée dans les assemblages haptiques du fait que :

- elle révèle de nombreux problèmes haptiques rencontrés dans les applications industrielles
- elle fait intervenir des objets simples à concevoir et permet d'élaborer des scénarii faciles à mettre en place ;
- elle parle naturellement à l'utilisateur (test psychomoteur basique chez l'enfant), quelque soit son niveau en RV ou CAO.

Elle nous permet, par ailleurs, de positionner nos résultats par rapport à ceux proposés dans la littérature pour le même type d'application. Nous décrivons, dans un exemple donné à la section 3.3.5, la modélisation de ce type d'assemblage ainsi que les types de guides géométriques à mettre en place pour réaliser l'insertion. Nous proposons, en outre, de valider la description du guidage telle qu'elle est présentée en section 3.3.7 par des graphes à états. La figure 4.7, représente différentes captures d'écran de notre application. Elle rappelle le mode normal

de fonctionnement du guidage, depuis la phase exploratoire (à gauche), jusqu'à la phase de guidage géométrique (au milieu) et la phase de guidage contraint (à droite).

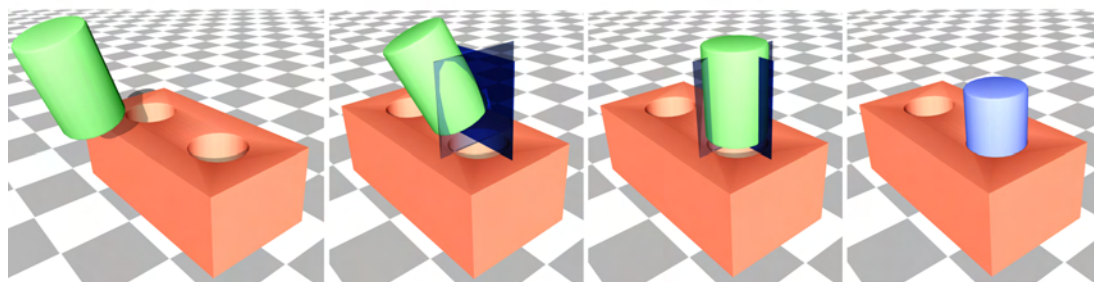


FIG. 4.7 – Le guidage virtuel contraint (VCG) appliqué, en situation, au *peg-in-hole*. De gauche à droite : le pion vert est positionné à la verticale au dessus du trou à l'aide du guidage géométrique, puis les collisions sont désactivées et la contrainte cinématique est appliquée (le pion devient bleu).

Pour notre manipulation, nous nous inspirons de l'expérience proposée par Amirabdollahian [AGJ05], qui propose une étude anthropocentrée des trajectoires de paires d'insertion *peg-in-hole*. Étant donné que son étude ne se place pas dans les mêmes conditions expérimentales que les nôtres, nous ne pourrions toutefois pas comparer nos résultats temporels avec ses travaux. Nous conservons l'idée des insertions enchaînées pour notre expérience, où nous simulons des enchaînements d'insertion entre 4 trous disposés en carré dans la scène virtuelle. L'intérêt d'effectuer la même tâche sur plusieurs objets est de réaliser plus rapidement les mesures et de donner une cohérence au scénario, plutôt que de recommencer n fois la même tâche. Nous définissons d'ailleurs 4 tâches d'insertion afin d'évaluer l'assemblage des objets :

- en situation de montages et démontages successifs ;
- selon que les utilisateurs sont droitiers ou gauchers (insertions à gauche ou à droite) ;
- selon la distance entre l'utilisateur et l'assemblage (insertions proches et lointaines).

Pour cette manipulation, les objets 3D que nous utilisons sont issus du logiciel Dassault CATIA. Ils sont premièrement conçus sur le modelleur CAO, puis exportés au format STL (maillage triangulaire). Ils sont alors convertis au format OBJ (maillage polygonal) par le logiciel Autodesk 3DSMAX. Des corrections mineures

sont réalisées sur les maillages (triangles inversés et trous dans le maillage), tandis que les tailles des polygones sont réglées les plus fines possibles (telles que proposées par les logiciels). Les défauts des géométries ainsi obtenues ne gênent pas notre manipulation, étant donné la manière dont le VCG traite les défauts géométriques des objets pour leur assemblage.

Le pion est généré par l'extrusion d'un profil circulaire. Le trou, quant à lui, est créé dans un bloc cubique, par une fonction de soustraction de volume ; ce volume est basé sur le même profil circulaire que celui du pion, nous assurant la similitude théorique des diamètres. Nous avons, par ailleurs, ajouté des congés de part et d'autre du cylindre pour simplifier son positionnement au-dessus du trou, sans changer le type de liaison simulée.

Pour notre manipulation, nous définissons manuellement le type de liaison à mettre en place pour chaque assemblage, et nous introduisons manuellement les données géométriques qui lui sont associées. C'est ainsi que nous modélisons chaque tâche d'insertion par une liaison pivot-glissant. L'axe de la contrainte cinématique qui modélise cette liaison est positionné colinéairement à l'axe du trou. Les caractéristiques géométriques du trou (cylindrique) sont préalablement récupérées dans le modèleur : l'orientation de l'axe du trou est donnée par la direction de l'extrusion et la position du centre du cercle de base est connue.

Le guidage que nous utilisons est de type *guide-guide* ; c'est-à-dire qu'un guide mobile est associé au pion et un guide fixe est associé au trou.

La géométrie du guide fixe est composée de deux surfaces planes carrées 2D qui s'intersectent en leur milieu (voir figure 4.8). Cette géométrie, que nous nommons "plans en croix" est positionnée au-dessus du trou de sorte que son axe soit colinéaire à l'intersection des deux plans (une droite). La hauteur de ces surfaces est égale à la hauteur du pion, et la largeur égale à (environ) une fois et demie le diamètre du trou. Ces deux dernières valeurs, qui ne font pas ici l'objet d'une étude approfondie, sont définies par notre expérience du *peg-in-hole*. Par ailleurs, nous posons l'hypothèse que les "guides plans" sont plus efficaces si l'approche du pion coïncide avec un quartier plutôt qu'avec une arête, surtout si ces quartiers sont entièrement visibles. Étant donné la position de la caméra, nous privilégions ainsi les enchaînements droite-gauche et devant-derrière, en orientant les guides selon les diagonales des blocs sur lesquels ils reposent (cas rose à droite sur la figure 4.8).

La géométrie du guide mobile est composée d'un segment. Ce segment est, en pratique, construit comme un cylindre très fin positionné sur l'axe central du pion. Cette géométrie est liée en mobilité au pion par une liaison encastrement. Notons que l'erreur spatiale entre l'axe théorique du pion et le diamètre du cylindre *quasi 1D* est non négligeable mais nécessaire étant donné la méthode de gestion des contacts que nous utilisons. On considère toutefois cette erreur comme acceptable dans le cadre de cette manipulation. Notons toutefois que pour une simulation

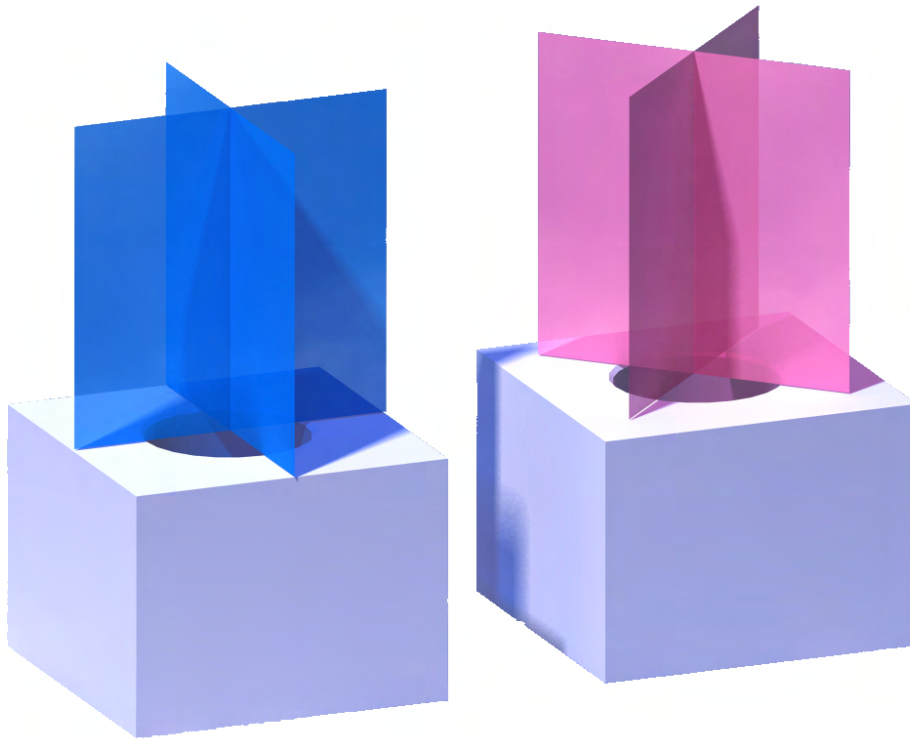


FIG. 4.8 – Deux différentes façons de positionner le guide fixe au dessus des blocs.

d'assemblage rigoureuse, cette erreur est corrigée une fois l'objet assemblé et détaché du périphérique haptique.

4.3.5 Description et spécificités des essais

Comme nous l'avons défini précédemment, l'utilisateur contrôle un pion via le périphérique haptique, et l'insère dans 4 blocs percés à tour de rôle. Les 4 blocs, A , B , C et D sont disposés en carré dans la scène. Ils sont fixes par rapport à l'environnement. La caméra est également fixe par rapport à l'utilisateur de sorte que le bloc A est face à lui au premier plan et que le bloc C est au dernier plan. Durant la manipulation, l'utilisateur enchaîne les tâches d'insertion entre les 4 blocs suivant des séquences prédéterminées, différentes à chaque manipulation. Les séquences sont définies aléatoirement, mais respectent les règles suivantes :

- pour chaque guidage à tester, on réalise 12 insertions, réparties uniformément sur les 4 blocs ;
- deux insertions consécutives ne se situent jamais au même endroit ;
- les 12 combinaisons d'enchaînement (AB, BA, BC etc. voir figure 4.9) sont présentées pour chaque type de guidage à tester.

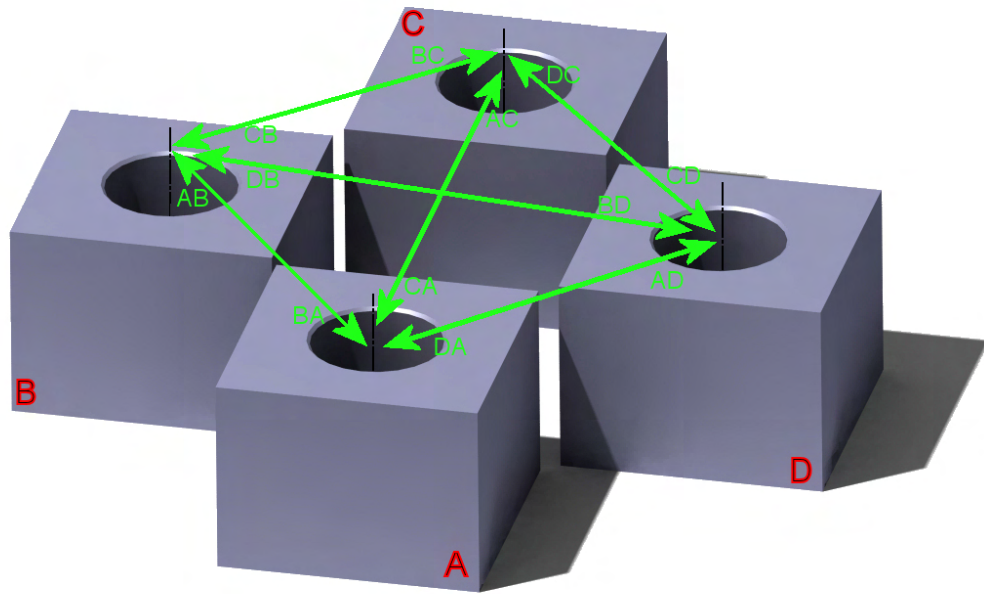


FIG. 4.9 – Différentes combinaisons de mouvement entre les 4 trous.

Le guidage géométrique ajoute des géométries fictives dans la scène. Du point de vue de l'utilisateur, ces géométries n'ont *a priori* pas de rôle dans l'assemblage et risquent de parasiter la compréhension de l'assemblage. Une attention toute particulière est donc donnée à l'utilisation de ces guides. Les métaphores à mettre en place, ainsi que les conditions d'application du guidage doivent donc permettre à l'utilisateur d'intégrer les guides géométriques au même titre que les objets à assembler. Deux cas d'interaction visuelle sont envisageables :

- le système de guidage géométrique peut être transparent pour l'utilisateur, c'est-à-dire que visuellement les géométries n'apparaissent pas à l'écran durant les phases exploratoire et/ou fonctionnelle ;
- le système de guidage géométrique est mis en évidence visuellement afin de signifier à l'utilisateur quel type de guidage il subit.

Nous souhaitons évaluer l'influence de différentes métaphores de guidage sur la performance de l'utilisateur. Les différents types de guidages, au nombre de 8, sont définis en fonction de la visibilité et de l'interaction haptique des guides géométriques. Les paramètres qui définissent chaque cas de guidage sont :

- l'activation ou non des contacts entre les guides géométriques ;

- la visualisation ou non du guide mobile (le segment) ;
- la visualisation ou non du guide fixe (les plans).

D'autres paramètres, que nous n'avons pas retenus dans le cadre de notre expérimentation, peuvent influencer les performances des utilisateurs. Le type de périphérique, par exemple, a une importance majeure sur les performances. Nous considérons ici un périphérique Virtuoso 6D35-45, qui propose des retours d'efforts 6D (3 translations et 3 rotations). Nous pourrions envisager, à la place, des périphériques à retour d'effort 3D tels que les Omni Phantom ou les interfaces haptiques de type Spidar. Les 3 paramètres que nous avons retenus définissent, par combinaisons, les 8 cas de guidage différents que nous testons. Les 8 cas sont les suivants :

	Contact guide-guide	Visu guide fixe	Visu guide mobile
Cas 1	0	0	0
Cas 2	0	0	1
Cas 3	0	1	0
Cas 4	0	1	1
Cas 5	1	0	0
Cas 6	1	0	1
Cas 7	1	1	0
Cas 8	1	1	1

La lecture de ce tableau nous permet, par exemple, de définir le cas de guidage numéro 6 : le guide fixe (les plans) est invisible mais interagit haptiquement avec le guide mobile (le segment), qui lui est visible.

Durant la manipulation, les différents cas de guidage sont imposés à l'utilisateur. Chaque cas est répété 12 fois, répartis sur les 4 blocs troués. Par exemple, le cas 3 est appliqué successivement sur les trous {D, A, D, C, B, A, C, D, B, C, A, B} puis le cas 4 est appliqué sur les trous {D, B, A, D, C etc. }. Afin d'éviter d'éventuels parasitages par des éléments extérieurs et homogénéiser la combinatoire, les différents cas de guidage sont présentés selon une méthode de permutation circulaire entre chaque sujet. Par exemple, le premier sujet suivra la séquence {cas 1, cas 2, cas 3, cas 4, cas 5, cas 6, cas 7, cas 8, cas 1}, le deuxième sujet la séquence {cas 2, cas 3, cas 4, cas 5, cas 6, cas 7, cas 8} et ainsi de suite. Chaque sujet effectue 8 séries de 12 essais non interrompues : une série correspond à une succession d'essais portant sur le même cas de guidage géométrique ; un essai correspond à une tâche d'assemblage du pion dans un trou (désinsertion $(n-1)$ + insertion (n)). Chaque sujet réalise donc $8 \times 12 = 96$ couples insertion / désinsertion. On choisit le facteur de déplacement du Virtuoso de sorte que l'utilisateur dispose d'une manipulation fine durant la phase de positionnement géométrique. Toutefois, la distance entre chaque trou, qui ne peut être mesurée

Pour chaque insertion, on active uniquement le VCG du trou cible sur lequel on souhaite effectuer une mesure. Les autres trous, incluant celui dont est extrait l'objet, constituent alors l'environnement fixe. Toutes les interactions entre le pion et ces trous sont considérés comme faisant partie de la phase exploratoire. Pour chaque tâche d'insertion, on mesure le temps d'accomplissement de l'utilisateur. Cette mesure est réalisée de manière automatique dans la simulation via une fonction horloge (*clock()* en langage C++). On mesure le temps entre le moment où le pion entre dans la zone fonctionnelle, *i.e.* quand les guides géométriques sont activés (position A sur la figure 4.10) et le moment où le pion est inséré dans le fond du trou (position B sur la figure 4.10). En procédant ainsi, on distingue clairement la phase d'assemblage de la phase exploratoire, sur laquelle on ne souhaite pas réaliser de mesure. Étant donné que les trous sont disposés en carré, cette distinction permet d'unifier les mesures, que le pion se déplace le long d'un coté du carré (ex. AB) ou le long d'une diagonale du carré (ex. AC). Grâce aux expériences préliminaires que nous avons réalisées (voir section 4.3.3), on s'assure également que les mesures ne sont pas influencées par les temps d'insertion au moment où les objets sont cinématiquement contraints.

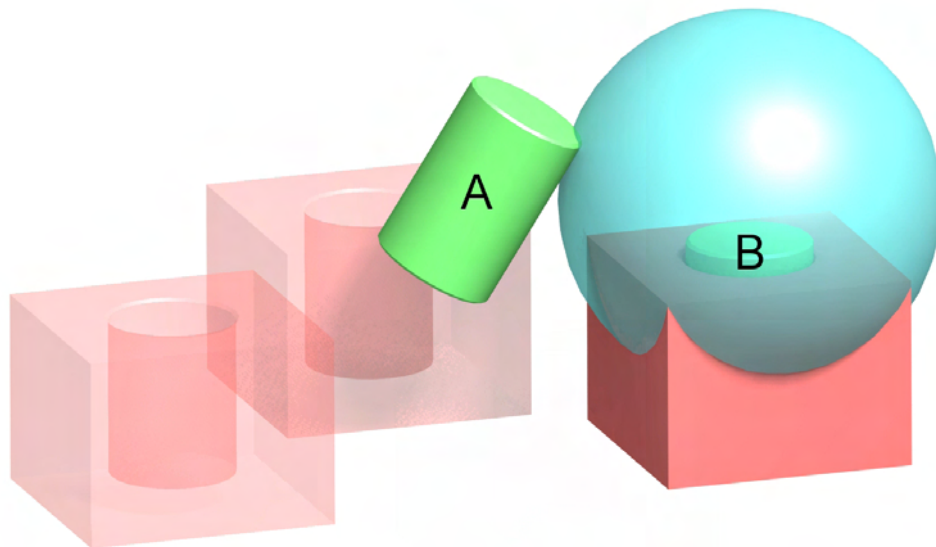


FIG. 4.10 – Les mesures temporelles sont réalisées entre le moment où le pion approche de la zone d’assemblage et le moment où le pion est au fond du trou.

4.3.6 Description d'un essai et des informations recueillies

Avant de débiter l'expérience, on demande aux sujets de se positionner de manière confortable, l'angle de la caméra étant donné et non modifiable. La durée totale d'une expérience pour un sujet est comprise entre 25 et 35 minutes, selon qu'il soit habitué ou non à la manipulation d'un *Virtuose*. Les sujets peuvent prendre une pause quand ils le souhaitent, pourvu qu'elle ait lieu juste après le succès d'un assemblage, c'est-à-dire juste après la fin d'une mesure. Au cours de la manipulation, on demande au sujet de décrire (sommairement) ce qu'il pense du cas de guidage en cours. Le questionnement porte uniquement sur l'appréciation qualitative du guidage et de sa comparaison avec le guidage précédent. On note toutefois toutes les remarques (à défaut de pouvoir enregistrer la manipulation). À l'issue de la manipulation, on fait remplir un questionnaire. Celui-ci requiert une évaluation subjective des différentes combinaisons de guidage qu'il a testées. On demande à l'utilisateur de noter, sur une échelle de 5 la facilité qu'il a eu à assembler l'objet.

Afin de prendre en main l'interface haptique et d'expliquer sommairement l'assemblage *peg-in-hole* qui suit, chaque sujet passe quelques minutes sur une démonstration "pré-manipulation". Cette démonstration présente trois tâches simples, non mesurées, et réalisées successivement :

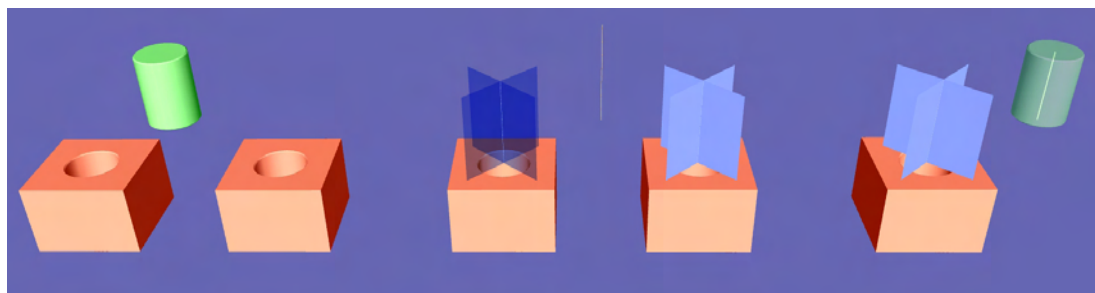


FIG. 4.11 – Démonstration préliminaire.

La première tâche consiste à manipuler haptiquement un pion pour l'insérer dans deux trous côte à côte. Dans cette démonstration, le diamètre du pion est diminué de sorte qu'il s'insère directement dans le trou sans aucun guidage. Le sujet s'entraîne alors à insérer le pion successivement dans les deux trous (paire d'objets à gauche sur la figure 4.11). La deuxième tâche consiste à manipuler un segment (quasi 1D) et à le faire glisser à l'intersection de deux plans. L'intersection de ces deux plans est une droite sur laquelle on demande à l'utilisateur de "coller" le segment qu'il contrôle (paire d'objets au milieu de la figure 4.11). La troisième et dernière tâche mixe les deux premières tâches. Elle consiste à positionner le

pion au-dessus du trou, sachant que l'on a activé les interactions entre le pion et le trou, et entre le segment et les plans (objets à droite de la figure 4.11).

Pour s'assurer que cette démonstration n'influe pas sur notre expérience, on utilise tous les paramètres (de visualisation et retour haptique), sans jamais les décrire aux sujets, et sur les trois démonstrations. L'expérience nous montre que cette phase d'apprentissage sommaire permet de pallier le problème de compréhension du guidage "guide-guide". En effet, les premiers essais qui ont été réalisés sans pré-manipulation, nous ont montré la difficulté de l'ensemble des sujets à comprendre et utiliser le système de guidage. Nous ne rapportons pas ces essais dans le cadre de cette étude.

Chaque tâche d'insertion peut être décomposée séquentiellement. Nous proposons une analyse de cette décomposition à la section 3.3.7, en présentant sous forme de grafcet les différentes étapes et transitions qu'un utilisateur doit respecter pour mener à bien l'assemblage. Nous vérifions expérimentalement que cette décomposition est bien justifiée à travers l'observation des méthodes d'approche et d'insertion des sujets. On leur propose alors le scénario suivant :

1. À l'issue de la pré-manipulation, l'utilisateur passe à la manipulation expérimentale. Celle-ci consiste en l'insertion du pion dans un des quatre trous, selon des séquences prédéterminées, mais dont l'utilisateur n'a pas connaissance. Chaque séquence met en scène des paramètres de guidage géométrique différents, mais toutes comportent le guidage contraint tel que nous l'avons modélisé.
2. Le pion est initialement positionné au milieu du carré formé par les 4 blocs. Le bloc cible, dans lequel l'utilisateur doit insérer le pion, change de couleur pour indiquer qu'il est la cible. La mesure temporelle est alors lancée lorsque le pion est suffisamment proche du bloc.
3. En fonction de l'activation des paramètres de guidage, l'utilisateur est alors "plus ou moins" guidé pour positionner le pion verticalement au-dessus du trou. Il dispose de toutes ses libertés de mouvement ainsi que du choix de l'approche, dans la limite des contacts inter-guides.
4. Une fois le pion en position, la contrainte lui est appliquée afin d'en restreindre les mouvements. Le pion change alors de couleur pour signifier qu'il est en mode contraint. Là encore l'utilisateur peut choisir d'extraire le pion, et dans ce cas il doit recommencer la phase d'approche. Il peut néanmoins finir la tâche d'assemblage en insérant le pion au fond du trou, où un dernier changement de couleur lui signifie la réussite de l'assemblage.
5. Une fois l'assemblage réalisé, un autre bloc est pointé comme nouvelle destination. L'utilisateur extrait le pion, toujours contraint, jusqu'à l'extraire

totale. La contrainte est alors désactivée et l'objet est de nouveau libre de mouvement.

4.3.7 Construction du plan d'analyse expérimentale

Quinze utilisateurs ont réalisé les expérimentations, dont une femme, âgée de 24 à 33 ans, parmi lesquels on compte un gaucher. L'ensemble des utilisateurs sont familiers des environnements 3D. 12 sont experts en réalité virtuelle dont deux spécialistes des interfaces à retour d'effort. Aucun utilisateur n'a de connaissance en guidage virtuel ou assemblage interactif.

Les analyses statistiques que nous décrivons ci-après s'appuient donc sur les résultats de $15 \times 96 = 1440$ mesures. Les facteurs d'analyse sont :

- le sujet ;
- la visualisation du guide fixe ;
- la visualisation du guide mobile ;
- l'activation haptique du guide fixe ;
- l'activation haptique du guide mobile ;
- la position de la cible par rapport à l'environnement ;
- la position de la cible par rapport à sa provenance.

Dans l'analyse qui suit, nous observons l'influence de ces facteurs sur les variables liées à la performance temporelle. Nous limitons nos analyses à l'étude des moyennes et des écarts-types sur les mesures temporelles pour chaque cas de guidage et pour chaque séquence d'assemblage. Ces analyses sont réalisées manuellement sur un tableur qui nous donne, pour chaque analyse statistique, un tableau qui montre l'évolution des variables testées.

Notre analyse donne des éléments de réponse à trois grandes catégories de questions :

- l'usage du guidage géométrique apporte-t-il un gain à l'utilisateur ?
- les métaphores visuelles associées au guidage géométrique aident-elle l'utilisateur ?
- y a-t-il une variabilité intersubjective importante ?
- y a-t-il un apprentissage de l'utilisateur au cours des différentes séries d'essais ?
- le système de guidage VCG apporte-il un gain en performance sur ce type d'assemblage ?

Pour les cas 1 à 8, on demande au sujet de manipuler les pions dans des conditions normales d'utilisation, c'est-à-dire sans préjuger de la vitesse ou de la précision des insertions. Les mesures étant réalisées au moment où le pion arrive en phase d'assemblage, on isole les temps liés aux déplacements entre chaque

zone ; d'autant que la distance entre les zones diffère selon que l'on considère les séquences A-C-A ou C-D-C (voir figure 4.9).

En général, une fois que l'utilisateur approche suffisamment le pion du trou cible (indiqué en bleu), déclenchant ainsi la mesure, il ne renonce pas à la tâche et persévère jusqu'au succès de l'insertion. Notons que ce succès est atteint pour l'ensemble des 1440 essais. Il est toutefois arrivé 9 fois que le pion ressorte du trou tandis que la contrainte cinématique était activée. Dans ce cas il était prévu que le pion doive être positionné de nouveau par l'utilisateur sans aucun guidage géométrique (cas d'assemblage 1).

Nous procédons également à l'évaluation subjective du guidage géométrique :

- en observant, pendant les manipulations, les mouvements et stratégies d'approche mises en place par l'utilisateur ;
- en recueillant, pendant les manipulations, les commentaires des sujets ;
- en exploitant les questionnaires soumis aux sujets à l'issue de chaque expérience.

4.3.8 Analyse des données expérimentales pour la performance

On constate que les sujets gagnent significativement en rapidité au fil des essais. La figure 4.12 montre que le temps moyen des utilisateurs diminue au fil des séries, et ce, indépendamment des séquences et des cas de guidage proposés. En moyenne pour l'ensemble des sujets, sur les 96 essais, ce temps passe de 14.53 s au début des manipulations à 8.75 s à la fin. L'écart-type des mesures diminue également de 9.94 s à 4.41 s (voir figure 4.13) pour l'ensemble de la population. Ces performances semblent évoluer selon un logarithme inverse. Cela peut s'expliquer par le fait qu'une fois l'effet d'apprentissage passée aux alentours de 40-50 essais, les performances restent stables dans le temps.

La figure 4.14 synthétise l'ensemble des performances des essais. Elle propose les moyennes des temps d'insertion, de tous les sujets, en fonction des cas d'assemblage (cas 1 à 8) et des positions des objets cibles (cibles A à D). Le temps moyen, pour l'ensemble des sujets et l'ensemble des essais est de 9.68 s. La figure 4.15 propose les moyennes des écarts-types selon le même ordonnancement. L'écart-type global, pour l'ensemble des essais et l'ensemble des sujets est de 5.40 s.

La moyenne des temps associée aux méthodes de guidage [1 ; 4] est de 10.64 s, contre 8.43 s pour les méthodes [5 ; 8], soit une différence de 20.8 %. Les écarts-types sont, pour cette comparaison, quasiment équivalents avec 5.35 s pour les cas [1 ; 4] contre 5.46 s pour les cas [5 ; 8]. Ces résultats montrent assez clairement que l'utilisation des guides est meilleure lorsque ceux-ci se comportent comme des

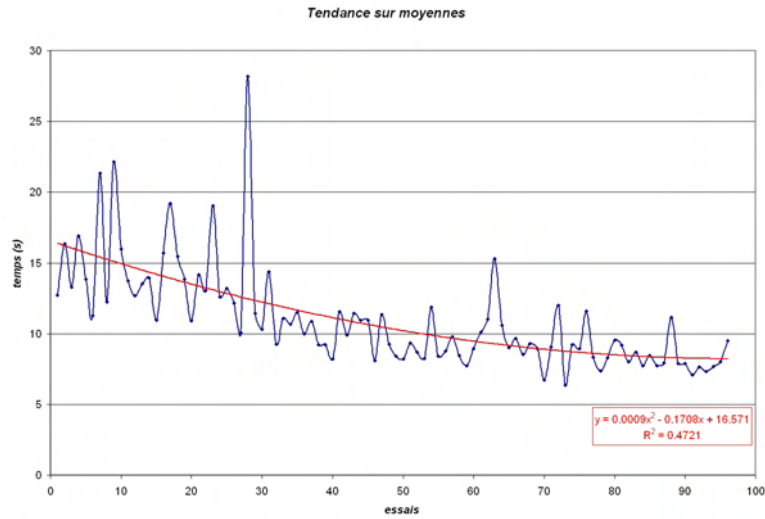


FIG. 4.12 – Amélioration des performances au cours des expériences, due à l’effet d’apprentissage.

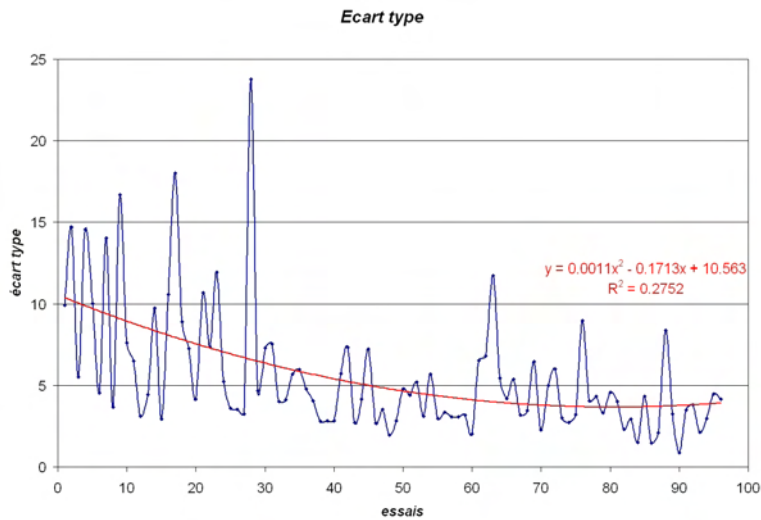


FIG. 4.13 – Convergence de l’écart-type, due à l’effet d’apprentissage.

surfaces impénétrables. En effet, nous confortons nos hypothèses sur la nécessité d’activer les collisions entre les guides géométriques en remarquant que, indépendamment des assistances visuelles proposées, toutes les moyennes du groupe [5 ; 8] sont inférieures à la plus petite des moyennes du groupe [1 ; 4].

Pour les cas [1 ; 4], on prend pour référence le cas 1, où l’utilisateur doit positionner le pion sans aucune aide (ni visuelle, ni haptique). L’assistance visuelle

Cas	1	2	3	4	5	6	7	8	Moyenne
Cible A	9.24	10.1	10.32	13	8.95	9.21	7.71	7	9.44
Cible B	9.13	9.26	11.17	13.4	7.37	9.28	7.72	7.27	9.44
Cible C	10.54	9.52	10.97	13.24	9.76	9.05	8.87	8.57	10.07
Cible D	9.92	8.56	11.29	10.57	9.06	8.77	8.07	8.25	9.65
Moyenne	9.71	9.36	10.94	12.55	8.79	9.08	8.09	7.77	9.68

FIG. 4.14 – Moyennes des temps d'insertion, pour l'ensemble des sujets, classées par cibles et cas d'assemblage.

apportée par la visualisation du guide fixe (cas 2) semble ainsi très légèrement aider au positionnement du pion. Toutefois, la visualisation du guide mobile (cas 3) fait diminuer les performances de l'insertion de 1.7 s. Les performances chutent même de 25 % quand les deux guides sont simultanément affichés (cas 4). Cependant, il faut noter que l'écart-type des performances pour le cas 4 est supérieur à celui des performances pour le cas 1. Ce résultat traduit une plus grande variabilité des comportements pour le cas 4, ce qui peut signifier que la diminution de performance peut masquer des résultats temporels dispersés.

Ces observations sont complètement opposées pour les cas [5;8], où l'ajout des visualisations des guides augmentent les performances, en particulier pour les cas 7 et les cas 8. Pour les cas [5;8], on prend pour référence le cas 5, où l'utilisateur doit positionner le pion sans aucune aide, ni visuelle, ni haptique. Si la visualisation du guide mobile ne semble rien apporter, 9.08 s contre 8.79 s, la visualisation des plans augment les performances du cas 8 %, et la visualisation simultanée des deux guides augmente les performances de l'ordre de 12 %. C'est ce cas 8, avec une moyenne de 7.7 s qui semble ainsi donner les meilleures performances. Nous confortons cette conclusion en observant les écarts-types de ces deux derniers cas, respectivement 4.35 s et 4.78 s, qui sont les plus serrés de l'ensemble des cas de guidage.

Cas	1	2	3	4	5	6	7	8	Moyenne
Cible A	3.25	6.64	6.4	4.41	6.99	6.46	4.14	3.95	5.28
Cible B	5.6	4.15	3.93	6.14	5.01	6.92	4.39	4.19	5.04
Cible C	5.21	4.75	6.55	9.43	6.72	6.14	4.79	5.7	6.16
Cible D	7.26	3.17	4.26	4.4	6.28	6.18	4.07	5.29	5.11
Moyenne	5.33	4.68	5.29	6.1	6.25	6.46	4.35	4.78	5.40

FIG. 4.15 – Moyennes des écarts-types, pour l'ensemble des sujets, classées par cibles et cas d'assemblage.

Afin de positionner nos résultats, nous pouvons étudier les travaux de Unger [UNT⁺02], qui évalue les performances spatiales et temporelles d'un utilis-

teur, pour une tâche d'insertion *peg-in-hole* carré. Bien que le cadre d'étude soit différent du nôtre : pion carré, pas de guidage, périphérique magnétique (etc.) les mesures temporelles réalisées sont proches des nôtres. En effet, elles mesurent le temps qu'il faut à un pion pour parcourir la distance entre un point de départ distant du trou, jusqu'à l'insertion complète au fond du trou. Cette étude de référence compare les trajectoires et les temps d'insertion pour les cas virtuels et réels, concluant que les stratégies de montage sont similaires dans les deux cas. Pour une manipulation d'insertion sans guidage ni retour d'effort, ce qui pourrait être l'équivalent de notre cas 1, ils obtiennent une moyenne de 12.3 s avec un écart-type de 1.55 s. Pour la même expérience avec un retour d'effort, ils obtiennent une moyenne d'insertion de 12.5 s (pour un écart-type de 3 s). Ils comparent également le temps d'accomplissement d'une tâche réelle, qui est bien inférieure : 4.1 s (écart-type : 1.54 s). En relativisant sur les conditions d'expérimentation, nous constatons que les performances de l'insertion du pion sont légèrement inférieures à celles d'Unger. L'écart-type est toutefois meilleur dans ces expériences, probablement du fait qu'ils réalisent les manipulations avec plus de sujets et pour moins de mesures.

4.3.9 Performance temporelle pour chaque cas de guidage

Les histogrammes présentés en figure 4.16 synthétisent la répartition des temps de mesure, en fonction du nombre d'essais pour chaque cas.

L'étude de ces histogrammes nous permet de mettre en évidence que les cas d'assemblage 5 à 8, qui proposent l'interaction haptique entre les guides, semblent jouer leur rôle pour la majeure partie des sujets. La concentration des mesures inférieures à 9.7 s (la moyenne totale) est proche de 60 % pour les cas [5 ; 8] contre 47 % pour les cas [1 ; 4]. De même la répartition des mesures est centrée en 5.5 s pour les cas [5 ; 8], alors qu'elle est de 9 s pour les autres cas.

On constate que les distributions des mesures suivent des lois normales (non centrées). Ce résultat était prévisible étant donné la répartition des niveaux de manipulation haptique de la population. En effet, le niveau d'expérience en haptique, obtenu par auto-évaluation des 15 sujets via un questionnaire, est réparti de façon suivante (figure 4.17) :

4.3.10 Analyse de l'influence de la localisation des cibles sur les performances

La caméra étant fixe dans l'environnement et les ombrages étant volontairement désactivés, chaque sujet doit se représenter la position des cibles par lui-même. Les facteurs d'influence que nous tentons d'extraire sont la position

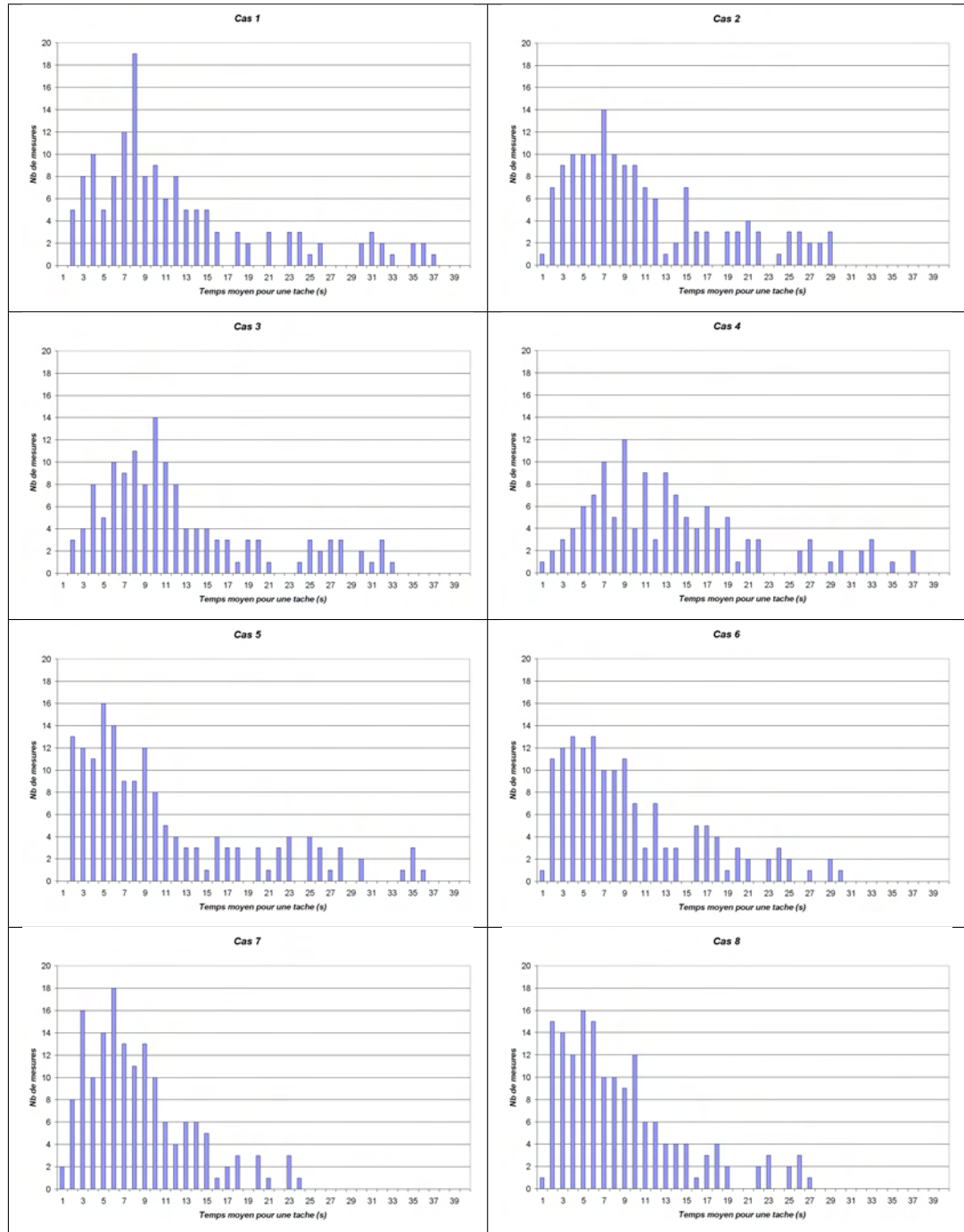


FIG. 4.16 – Répartition des performances temporelles, pour les cas 5 à 8

de l'assemblage cible par rapport à l'utilisateur, ainsi que le séquençage des insertions. Ce dernier point ne faisant pas partie du sujet initial, nous concluons

Estimation (subjective) sur une échelle de 5	effectif
Expert (5)	2
Connaisseur (4)	1
Initié (3)	7
Débutant (4)	3
Nulle (1)	1

FIG. 4.17 – Estimation (subjective) du niveau de manipulation haptique

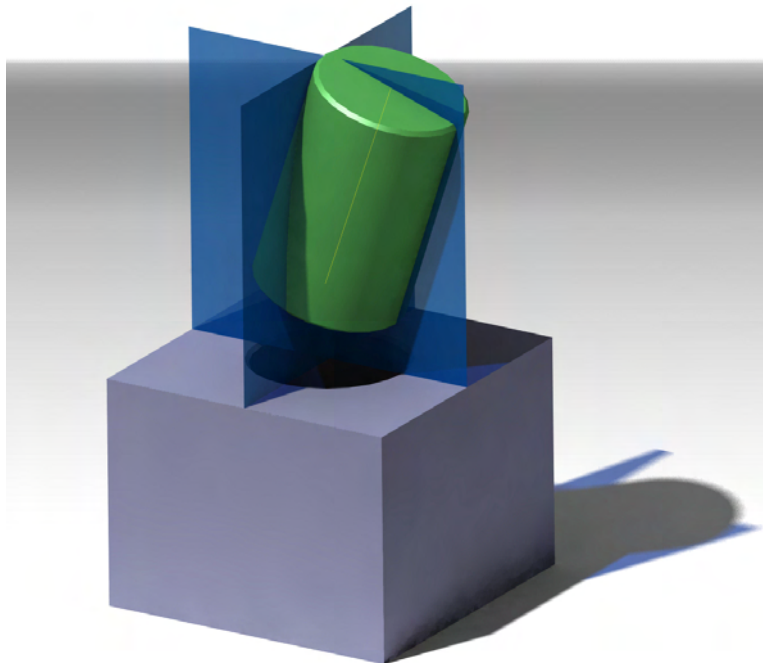


FIG. 4.18 – Le cas 8, qui propose l'interaction haptique et la visualisation des deux guides, offre les meilleures performances temporelles.

rapidement sur ce point. L'idée principale est d'évaluer le temps de positionnement des guides géométriques en fonction de :

- la localisation de la cible ;
- la séquence d'assemblage, c'est-à-dire le vecteur donné entre la provenance et la destination d'un objet ;
- le mouvement d'approche d'un pion vers sa cible.

Les moyennes temporelles pour chaque cible sont présentées à la figure 4.15. Nous observons, en premier lieu, que les sujets mettent en moyenne 9.51 s pour réaliser les assemblages A, B et D (écart-type faible de 0.12 s) et mettent 1/2

seconde de plus pour réaliser l'assemblage C. On explique facilement cette différence du fait que cette cible est lointaine dans la scène. On peut recouper cette conclusion avec le fait que le plus mauvais temps réalisé est obtenu en C, pour le cas 4, dont on a identifié qu'il offrait le guidage le moins performant.

Séquence	Temps (s)
A->B	10.22
A->C	10.38
A->D	9.17
B->A	9.98
B->C	9.53
B->D	10.3
C->A	7.54
C->B	8.9
C->D	7.78
D->A	10.10
D->B	10
D->C	11.03

FIG. 4.19 – Estimation (subjective) du niveau de manipulation haptique.

Le tableau 4.19 donne les moyennes des temps d'insertion en fonction des séquences d'assemblage du pion. La lecture de ce tableau indique, par exemple pour la première ligne, que le pion met en moyenne 10.22 s pour aller de la cible A à la cible B. Une lecture rapide de ce tableau nous montre des inégalités flagrantes entre les séquences. Les chiffres apparaissent par séries de trois, qui correspondent au point de départ de chaque cible. C'est ainsi qu'il faudra 9.93 s pour effectuer un assemblage au départ du point A et B. En revanche, il faut 8.7 s pour les assemblages au départ de C contre 10.8 s au départ de D. Ces deux derniers résultats sont plus parlants. On peut expliquer cette différence de 19.5 % du fait de l'interface Virtuose. En effet le groupe musculaire utilisé pour réaliser un mouvement qui suit l'axe des profondeurs, est sollicité différemment selon que l'utilisateur pousse ou tire la poignée de préhension. Si on compare deux à deux les trajectoires opposées, il ressort que les couples A-C et C-A offrent en effet une différence de temps significative :

A-B ; B-A = 0.24 s ;
 A-C ; C-A = 2.84 s ;
 A-D ; D-A = 0.93 s ;
 C-B ; B-C = 0.63 s ;
 B-D ; D-B = 0.3 s ;
 C-D ; D-C = 0.73 s.

D'un point de vue quantitatif, il apparaît que les sujets obtiennent des performances assez similaires. Les moyennes des écarts montrent qu'il y a finalement peu de différence entre les sujets, tandis que l'on constate de réelles différences entre les cas de guidage. Pour être plus précis, le niveau d'expertise (novice, intermédiaire ou expert) influence nettement les performances. En conclusion de ce paragraphe, nous pouvons affirmer que la distance a une influence importante sur le guidage. Si cette hypothèse paraît triviale, nous avons pu confirmer expérimentalement que l'utilisation des guides, et particulièrement leur visualisation est corrélée à la distance entre l'utilisateur et la cible. Notons que nous n'avons pas ici étudié l'influence d'un affichage stéréoscopique.

Analyse subjective de l'expérience et du système de guidage VCG Au regard des gains en performance temporelle que nous venons d'évaluer, nous souhaitons évaluer l'utilisabilité subjective du guidage. Pour cela nous effectuons des observations et nous questionnons les sujets au cours de la manipulation. Cette interaction peu intrusive avec les sujets permet de compléter l'analyse des questionnaires d'évaluation. Nous leur demandons seulement de commenter spontanément l'aisance qu'ils ont à chaque cas de guidage et s'ils le préfèrent par rapport au précédent. Étant donné que les cas d'assemblage apparaissent selon une méthode de permutation circulaire, on déduit facilement un ordonnancement qualitatif des différents cas (du moins préféré au plus préféré) : cas 1, cas 4, cas 2, cas 5, cas 6, cas 3, cas 7, cas 8. On remarque l'importance de la visualisation des plans, laquelle peut s'avérer plus pertinent que le guidage sans assistance visuelle. Nous résumons ci-après les remarques les plus fréquentes fournies par les utilisateurs :

1. le cas 1 est globalement le plus difficile (pour 13 sujets) ;
2. le cas 1, même s'il conduit à un "tâtonnement" du pion au-dessus du trou plus ou moins long, est tout de même facilement réalisable (pour 10 sujets) ;
3. le cas 8 est unanimement le plus apprécié (pour 15 sujets) ;
4. les utilisateurs adoptent des stratégies d'approche différentes en fonction des cas (pour 8 sujets) ;
5. l'apport de l'interaction haptique entre les guides est unanimement préféré (cas [5;8]) (pour 14 sujets) ; même s'ils nécessitent un temps d'adaptation (pour 8 sujets) ;
6. une fois coutumier du guidage haptique, l'insertion est "immédiate" car il n'y a qu'à "faire simplement glisser" le pion dans le trou (pour 11 sujets) ;
7. dans les cas [5;8], la visualisation des guides aide à la compréhension du système de guidage et change la stratégie d'approche ; les plans aident à situer l'axe du pion dans la scène par rapport aux "quartiers" formé par les plans (pour 13 sujets) ;

8. lorsque les interactions entre guides sont désactivées (cas [1;4]), la plupart des utilisateurs se basent sur les géométries des guides pour positionner le pion, plutôt que sur les interactions pion-trou (pour 9 sujets) ;
9. lorsque les cas 5, et 6 sont présentés en premier, l'utilisateur peut avoir des difficultés à comprendre d'où vient la "contrainte" qu'il ressent (pour 4 sujets) ;
10. lorsque le cas 0 est présenté en premier, les utilisateurs préfèrent successivement chaque nouveau guidage à l'ancien (pour 2 sujets). L'inverse n'est pas vrai lorsque le guidage 8 est présenté en premier (pour 1 sujet).
11. pour la série [5;8] l'apport de la visualisation du guide mobile est minime (cas 6), sauf si elle est couplée à la visualisation des plans (cas 8) (pour 6 sujets) ;
12. les informations visuelles aident globalement plus qu'elles ne gênent, mis à part le cas 4-C où la somme des transparences et la distance rendent la visualisation très difficile (pour 10 sujets) ;
13. l'assemblage est globalement plus difficile au fond (pour 10 sujets).
14. certains utilisateurs ont régulièrement des difficultés avec le pion de gauche (resp. droite)(pour 6 sujets) ;
15. pour un cas donné, certains utilisateurs ne comprennent pas pourquoi ils arrivent plus difficilement à insérer le pion dans un trou plutôt qu'un autre (pour 3 sujets) ;
16. Les utilisateurs souhaitent globalement avoir des informations visuelles de profondeur ; l'affichage des plans les aident en ce sens (pour 9 sujets) ;
17. les plans doivent rester transparents plutôt qu'opaques. La transparence du pion, elle, semble inutile du moment que son axe est visible (pour 12 sujets) ;
18. l'expérience est quelque peu fatigante, dans la mesure où elle fait intervenir de nombreux décalages de repères (*i.e.* recentrage de la poignée du Virtuose au centre de l'espace de travail)(pour 3 sujets).

Notons également que pour la totalité des utilisateurs, l'application de la contrainte cinématique, une fois les objets positionnés, est complètement transparente. Les novices ne perçoivent d'ailleurs pas du tout que les mouvements du pion sont contraints une fois qu'il est inséré. Pour rendre compte aux sujets que l'on change le mode de contrôle, on leur demande d'essayer d'insérer le pion dans un trou non ciblé. Le guidage VCG y étant inactif, le pion ne rentre jamais dans le trou à la surprise des sujets. L'extraction, quant à elle, est légèrement perçue étant donné que le seuil de désactivation du guide est situé très légèrement au-dessus du trou. Cependant la quasi totalité des sujets ne perçoit pas le retour au mode dynamique non contraint.

En complément de ces évaluations subjectives, on réalise des observations sur les stratégies d'approche du pion par rapport au trou. Pour cela on observe les trajectoires suivies par le pion entre le moment où il entre dans la zone d'assemblage et celui où il est convenablement positionné. On observe également les différentes stratégies de positionnement en fonction des cas de guidage :

Quand les interactions haptiques entre guides sont désactivées, le sujet suit deux stratégies distinctes :

- une fois que le pion est au-dessus du trou, l'utilisateur se concentre sur les géométries du pion et du trou, indépendamment des métaphores graphiques des guides. Probablement influencé par les exemples de la pré-manipulation, il se guide haptiquement sur les congés des objets, par tâtonnement, jusqu'à ce que le pion entre dans le trou.
- l'utilisateur fait abstraction du contact entre le pion et le trou, pour se concentrer sur les indices graphiques. C'est-à-dire qu'il maintient le contact [base du pion, haut du trou] en essayant de trouver des stratégies basées sur des conditions géométriques. Par exemple, dans le cas 2 il tente de positionner cet axe le plus vertical possible ou tente de le rendre parallèle au arêtes du bloc. Dans le cas 3 il essaie d'égaliser les aires du cylindre coupé par les plans. Dans le cas 4, il essaie de faire coïncider les deux axes visibles.

Quand les interactions haptiques entre guides sont activées, les stratégies d'approche du pion sont les suivantes (voir figure 4.20) :

- le pion arrive de sorte que son axe est approximativement parallèle à la droite intersection des guides ;
- le pion arrive incliné dans le quartier formé par les plans ; dans ce cas la mise en correspondance des lignes directrices suit l'approche proposée au chapitre 3.3.5 ;
- le pion arrive par dessus les guides plans. Ce cas est le plus contraignant puisque, par réaction entre le segment et les arêtes supérieures des plans, le pion se positionne dans un des quartiers formés par les plans. Selon la localisation du trou et les métaphores graphiques utilisées, le sujet peut avoir des difficultés à comprendre où se situe le pion.

À l'issue des manipulations, nous demandons aux sujets de remplir un questionnaire. Nous leur demandons d'évaluer subjectivement, en notant sur des échelles de 5 points, les différents cas de guidage qu'ils ont testés. Nous leur demandons en particulier d'évaluer la facilité qu'ils ont ressentie à assembler le pion dans le trou. Nous avons collecté les différentes réponses des sujets, et réalisé des moyennes sur chaque question. Pour l'ensemble des sujets, les résultats sont synthétisés dans le tableau 4.21 suivant :

- l'échelle de "compréhension" est donnée : 1-pas comprise, 5-comprise ;

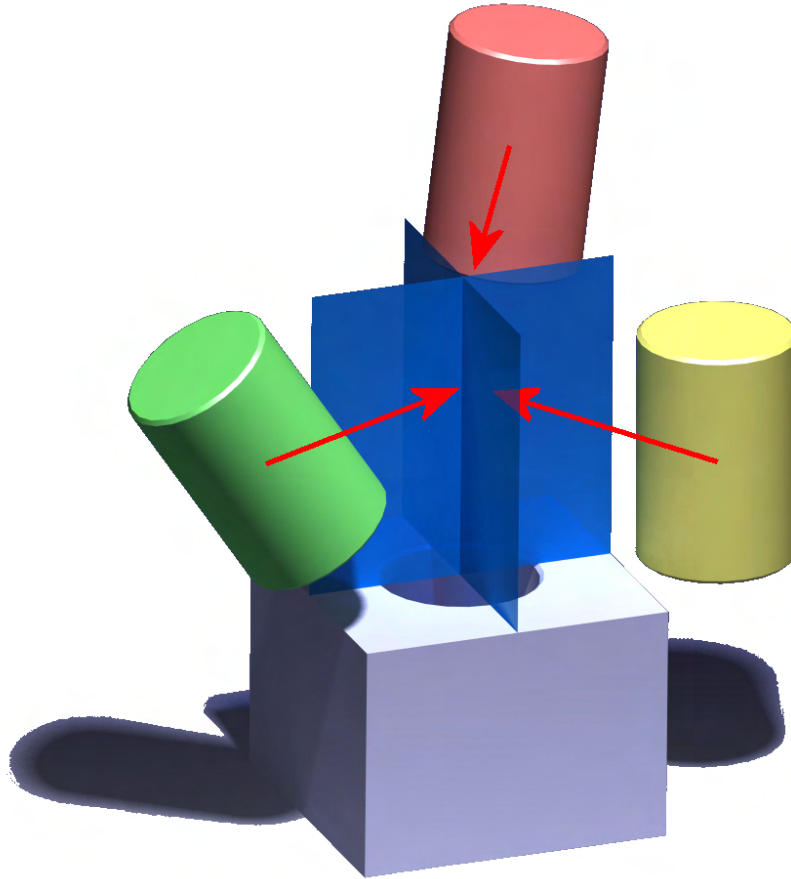


FIG. 4.20 – Moyenne des performances, en fonction des trajectoires du pion.

- l'échelle de "évolution" est donnée : 1-pas évolué, 5-évolué ;
- l'échelle de "facilité" est donnée : 1-difficile, 5-facile ;
- l'échelle de "fatigue" est donnée : 1-pas fatigant, 5-fatigant ;
- l'échelle de "réalisme" est donnée : 1-pas réaliste, 5-réaliste.

Ces données nous permettent d'identifier clairement l'apport de l'interaction haptique "guide-guide" pour les trois cas de visualisation [plan, segment, plan+segment] ; pour ces trois conditions, l'évaluation subjective nous montre un accroissement de 0.6 point entre l'utilisation non haptique des guides et l'utilisation haptique des guides.

Question	Réponses (note sur 5)
Compréhensions du système de guidage	4.4
Evolution estimée des performances	3.8
Facilité à insérer sans aucun guidage	1.8
Facilité d'insertion avec les plans, sans contact	2.2
Facilité avec les plans, avec contact	2.8
Facilité avec le segment, sans contact	2.4
Facilité avec le segment, avec contact	2.9
Facilité avec le plan et le segment, sans contact	2.4
Facilité avec le plan et le segment, avec contact	3
Fatigue	2.6
Réalisme du meilleur assemblage guidé	3.4

FIG. 4.21 – Traitement des données du questionnaire

4.3.11 Bilan des expériences

Un retour d'expérience rapide nous permet de constater que les sujets ont généralement peu de problèmes pour accomplir les essais et adoptent des comportements et des stratégies différentes, qui évoluent en fonction du temps et du type de guidage qu'ils comprennent. Même si des temps de réalisation aberrants apparaissent (ceux qu'on considère par exemple supérieurs à 15 secondes), ils restent marginaux. Cela montre que le cas 7, sur lequel on a ajouté des informations, les *virtual fixtures* apportent réellement une aide, y compris aux personnes qui sont des habituées des environnements virtuels et qui, peut-être, peuvent mieux gérer les différents problèmes liés à la perception de l'assemblage et des techniques d'approche.

Dans l'ensemble, les analyses quantitative et qualitative nous ont montré que la meilleure combinaison de guidage correspond à la visualisation simultanée des guides fixes et mobiles, et dont les collisions ont été activées (cas numéro 8). L'influence de la position de l'assemblage dans la scène semble aussi avoir une influence sur les performances des utilisateurs. L'insertion en arrière plan montre le plus de réticences et les plus mauvais résultats temporels, en conséquence de quoi les séquences "premier plan - arrière plan" sont unanimement les plus difficiles à réaliser.

On notera ensuite qu'il y a une certaine décorrélation entre les ressentis des utilisateurs concernant leur performance, et leurs performances réelles. En effet s'ils affirment préférer les modes de guidage [contact + visualisation], leurs performances temporelles ne reflètent pas vraiment cet écart. On peut en conclure un apport cognitif important, donné par les contacts "guide-guide" et par leur visualisation.

Enfin, pour compléter les résultats, on peut regarder l'influence du degré d'expertise sur l'apprentissage. On constate que, pour les différents cas de guidage, et pour toutes les séries, les experts dépassent les novices. Ensuite, la performance moyenne des niveaux de connaissance ["experts" + "connaisseurs"] semble bien meilleure que la performance du groupe ["initiés" + "débutants"]. Cependant, l'écart entre les résultats moyens de ces deux groupes a tendance à se réduire au fil des essais. La combinaison de la figure 4.12 et des résultats du tableau 4.14 va plus loin et montre que tous les sujets ont vu leur performance moyenne s'améliorer en utilisant les guidages [5 ;8] par rapport aux [1 ;4]

Gageons enfin que ces résultats s'appliquent pour une tâche très spécifique, le *peg-in-hole*, qui est répétée maintes fois au cours d'une même manipulation. Si l'effet d'apprentissage et le niveau de connaissance initial ne peuvent être négligés, nous devons extrapoler sur l'utilisation générale du système de positionnement géométrique "guide-guide", sans l'expérimenter sur un cas industriel unique. Dans la section suivante, nous présentons ainsi plusieurs démonstrations qui ont été réalisées sur la plate-forme VCG qui présente des applications industrielles réelles ainsi que des manipulations plus complexes des VCG.

4.4 Démonstration

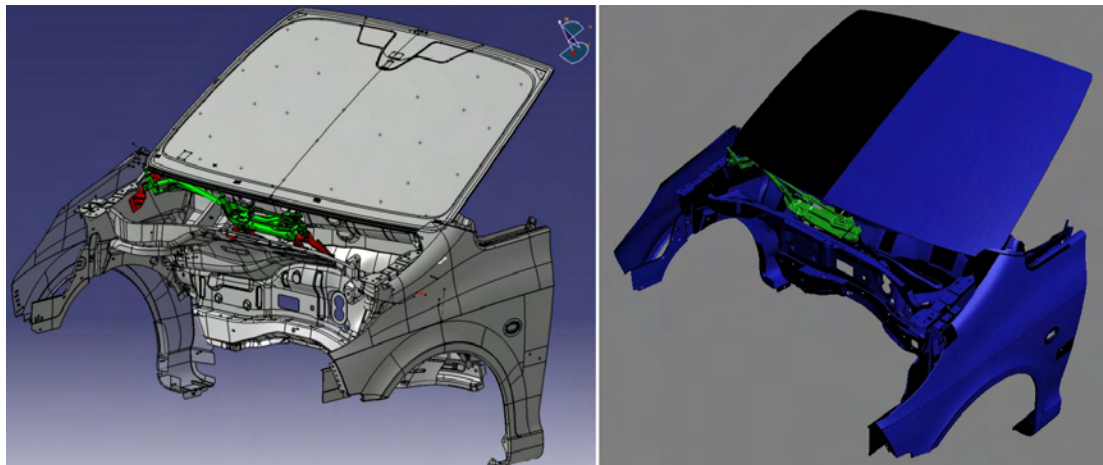


FIG. 4.22 – Montage d'un moteur d'essuie-glace : prétraitement des données 3D. Courtoisie de Renault.

La figure 4.23 illustre un cas industriel utilisant deux VCG pour l'assemblage d'objets CAO complexes. Il s'agit d'un assemblage "large" d'un moteur d'essuie-glace (objet vert sur la partie droite de la figure 4.23 dans un châssis d'automobile (objets rouges et jaunes sur la figure). Le moteur est contrôlé haptiquement selon

6 DDL. Cette application met en oeuvre des géométries au format STL issus d'objets CAO (CATIA). L'assemblage du moteur d'essuie-glace est défini comme un couple d'insertions simples de type pivot glissant aux deux extrémités de l'objet. Ce cas, que l'on peut considérer comme un double *peg-in-hole*, utilise deux VCG simultanés, qui permettent d'insérer et ajuster les objets. En tant qu'extension du cas simple présenté en section 3.3.7 les deux couples de guides géométriques sont construits avec des squelettes linéaires mobiles et des bi-plans fixes, qui offrent les mêmes critères d'activation des contraintes pivot-glissant, une fois que les deux pions du moteur sont bien alignés aux deux trous du châssis (figure 4.23).

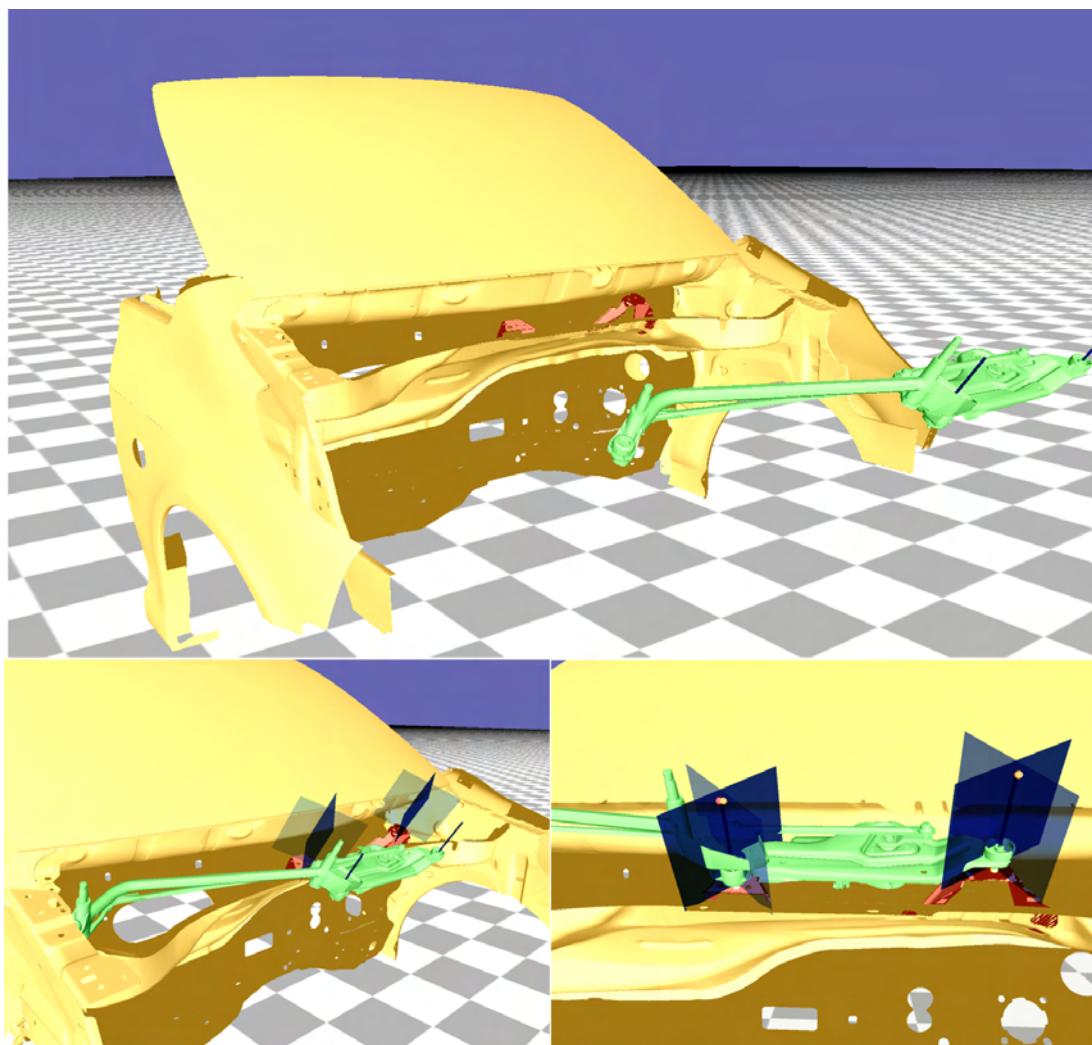


FIG. 4.23 – Implementation de la méthode VCG sur un cas industriel : l'assemblage d'un moteur d'essuie-glace (objet vert) dans le châssis d'une automobile.

4.5 Conclusion

Dans un premier temps, nous avons montré la faisabilité du couplage entre l'approche *time stepping* de la NSCD avec l'interaction haptique. Les performances temporelles du simulateur physique de Conf&is, évaluées en fonction du nombre de contacts gérés simultanément, nous encouragent à améliorer notre plate-forme, notamment en y adjoignant un détecteur de collision qui gère les modèles maillés complexes, et en continuant nos analyses de stabilité avec ce genre d'objets 3D. Les deux exemples que nous avons proposés nous ont permis d'évaluer subjectivement la qualité du rendu haptique en fonction des paramètres de la scène virtuelle. Dans le cadre des assemblages d'objets CAO, nous reconnaissons les possibilités de la dynamique non régulière haptique, pour l'exploration de scènes virtuelles dans une phase de pré-assemblage.

Pour les phases d'assemblage fonctionnel, nous avons montré la pertinence de notre méthode de guidage VCG pour des tâches d'insertion, et plus particulièrement les tâches modélisables en des liaisons pivot-glissant. Nous avons mis en lumière le gain en performances temporelles de l'utilisateur, en terme de temps d'accomplissement de tâche, sur le cas du multi *peg-in-hole* et par extension des tâches d'insertion. L'analyse subjective de l'utilisabilité nous a montré l'importance de la compréhension du fonctionnement du guidage et des métaphores à utiliser pour faire comprendre le guidage géométrique contraint, à l'utilisateur. Nous avons par ailleurs analysé les effets d'apprentissage du guidage sur une tâche répétitive, qui nous a permis de constater une amélioration réelle des performances de l'utilisateur avec notre méthode.

Nous reconnaissons que nos résultats s'appliquent pour des conditions très spécifiques d'assemblage CAO, avec une tâche d'insertion unique et canonique. Dans un souci de généricité, nous prévoyons de réaliser ce genre d'expériences pour d'autres types de modélisation VCG, par exemple des tâches de positionnement par mouvement hélicoïdal ou suivant des appui plan bilatéraux.

Conclusion générale et perspectives de recherche

Dans cette thèse, nous nous sommes intéressés au traitement des interactions entre objets 3D, dans le cadre spécifique des assemblages haptiques. L'objectif de la thèse est d'étudier les différents modes de dynamique applicables dans les simulations d'assemblages CAO. Ce domaine de recherche comporte plusieurs verrous technologiques et scientifiques concernant le couplage des interfaces haptiques avec des moteurs de simulation dynamique. La contrainte de temps interactif, très restrictive, impose à toute méthode dynamique de détecter des collisions et de calculer des efforts de contact de manière très rapide et avec des critères forts de stabilité mécanique. Dans les simulations d'assemblage, nous avons identifié deux phases de manipulation où le traitement dynamique des contacts peut être réalisé : soit de manière contrainte pour aider les utilisateurs à positionner des objets dans l'espace, soit de manière non contrainte pour obtenir un rendu haptique réaliste des interactions. L'approche proposée a consisté à séparer les interactions liées à l'exploration des scènes haptiques, des interactions liées aux assemblages. Au sein d'une même simulation d'assemblage, nous avons proposé d'utiliser des modèles de traitement dynamique non contrainte pour les phases exploratoires et l'utilisation de modèles de dynamique contrainte pour réaliser les assemblages haptiquement.

Dans un premier temps, nous avons rappelé les fondements de la manipulation haptique d'objets CAO, dans le cadre précis des applications de réalité virtuelle. Nous avons décrit les architectures des interfaces haptiques, avec lesquelles les utilisateurs réalisent des assemblages CAO. En plus des contraintes temporelles fortes imposées aux simulations, nous avons rappelé les contraintes haptiques sur la qualité du rendu haptique et sur la stabilité. Au sein des simulations interactives, nous avons présenté un bref état de l'art du couplage entre les systèmes haptiques et les simulations physiques, à partir duquel nous avons positionné notre propre couplage haptique dans la plate-forme Conf&tis. Pour les simulations haptiques d'assemblages CAO, l'identification de ces contraintes nous permet de poser un certain nombre de choix d'architecture et de type de couplage, afin d'obtenir des rendus haptiques satisfaisants pour les tâches d'assem-

blage. En recentrant ces problématiques du rendu haptique, nous avons proposé de dissocier les tâches d'assemblage en deux parties. La première se concentre sur la manipulation haptique "classique" d'objets 3D, où l'exploration haptique des environnements est primordiale. La deuxième se concentre sur les tâches d'assemblage d'objets complexes, où l'utilisateur doit réaliser des opérations précises et prévisibles sur les objets CAO qu'il manipule.

Dans le cadre de la phase exploratoire de l'assemblage interactif, la description des méthodes de traitement des interactions nous a rappelé, dans le deuxième chapitre, les problèmes liés à la détermination des efforts de contact entre objets. Nous avons détaillé l'approche par dynamique pénalisée ainsi que celle par dynamique non régulière, que nous avons considéré comme un modèle mécanique plus fidèle à la réalité. Nous avons identifié l'approche pénalisée comme étant la plus largement utilisée dans le domaine haptique grâce à l'aisance d'implémentation et la pertinence des résultats temporels. Nous avons alors détaillé les modèles non réguliers, étudiant les approches pilotées par les événements, qui imposent une contrainte au niveau du contact une fois que celui-ci a eu lieu. Nous avons finalement détaillé le modèle par temps contrôlé, qui prend en compte des lois comportementales tout en assurant un écoulement continu du temps. Nous avons enfin présenté l'implémentation de l'approche *time-stepping* au sein d'un simulateur haptique. Nous avons présenté notre plate-forme expérimentale Conf&tis. Ce prototype nous permet de valider la faisabilité du couplage entre un modèle mécanique de dynamique non régulière et l'interface haptique, et ce, en respectant les contraintes de temps réel et de rendu haptique.

Dans le troisième chapitre, nous avons proposé une nouvelle méthode qui utilise une combinaison de contraintes cinématiques et de guides géométriques virtuels, dérivés des *virtual fixtures*, pour réaliser l'assemblage d'objets CAO en environnement RV. En modélisant une tâche d'assemblage comme une combinaison de liaisons mécaniques dans l'espace fonctionnel d'assemblage, nous nous sommes intéressé au positionnement des objets et à l'activation de contraintes d'assemblage en temps réel. Nous avons, pour cela, décrit les difficultés liées à l'assemblage haptique d'objets CAO dans les simulations de RV. En introduisant le concept de guidage virtuel contraint, nous avons présenté une nouvelle méthode de guidage haptique qui décompose une tâche d'assemblage en deux étapes indépendantes : une étape de guidage qui emploie des géométries additionnelles en tant que *virtual fixtures* pour positionner des objets, et une étape d'assemblage qui emploie des contraintes cinématiques pour effectuer la tâche. Nous avons basé notre présentation sur l'exemple classique de l'insertion *peg-in-hole*, sur lequel nous avons réalisé une partie de nos expérimentations.

Dans le dernier chapitre nous avons présenté des expériences dont le but était de vérifier les gains temporels obtenus par les deux modèles dynamiques proposés. À travers Conf&tis, nous avons voulu montrer, dans la phase exploratoire,

l'intérêt du traitement des interactions par dynamique non régulière. La manipulation haptique d'objets simples nous a permis de tester le chargement de notre moteur physique, en terme de nombre de contacts. À travers deux expériences représentatives, nous avons évalué les performances de nos algorithmes et identifié les limites du rendu haptique. Dans un deuxième temps, nous avons montré l'apport de notre méthode de guidage pour les tâches d'insertion. Nous avons mis en place, de manière rigoureuse, des manipulations basées sur des enchaînements d'insertion *peg-in-hole*. Après avoir isolé les phénomènes d'apprentissage, nous avons clairement identifié les types de métaphores à mettre en place pour les guides géométriques "guide-guide". En plus de vérifier expérimentalement le gain en performance apporté par l'utilisation de notre guidage, nous avons proposé une brève analyse de l'utilisabilité de notre méthode. L'évaluation subjective des VCG nous a conforté dans le choix du mode de guidage contraint, et a corroboré les résultats quantitatifs.

Perspectives Nous espérons que les résultats de nos travaux vont permettre de renforcer l'utilisabilité de la réalité virtuelle pour la simulation des tâches d'assemblage, et ainsi faciliter la diffusion de cet usage dans l'industrie. Nous reconnaissons que la manipulation haptique d'objets CAO, dans ce contexte, se base essentiellement sur les aptitudes et le confort de l'utilisateur. Tout au long de cette thèse, c'est autour de l'utilisateur que nous avons concentré nos travaux, avec le souci permanent de l'assister opérationnellement. Dans la mise en place de notre méthode de guidage, ainsi que dans nos expérimentations, nous avons pris soin de vérifier qu'il garde un contrôle naturel de ses mouvements, condition nécessaire pour toute "bonne" interaction haptique.

Dans cette thèse nous avons essentiellement posé les bases méthodologiques de ce type de guidage et nous avons étudié des cas canoniques qui ont permis de valider les modèles. Le côté novateur de la méthode VCG ouvre ainsi la porte vers de nombreuses améliorations.

Nous avons suggéré, à la section 3.3.1 d'utiliser les informations de contact entre les objets assemblés, pour définir des liaisons cinématiques. La méthode de Jacob *et al.* [IML07] peut ainsi nous aider à automatiser la mise en place des VCG dans la phase de prétraitement, là où nous définissons manuellement les géométries et contraintes de guidage. La prochaine étape de notre travail consistera donc à intégrer ces travaux et vérifier leur pertinence pour la création et la mise en position des guides géométriques et des contraintes cinématiques.

Tout au long de ce manuscrit, nous avons illustré nos concepts sur l'exemple de l'insertion *peg-in-hole*. Ce cas canonique nous a permis de valider nos hypothèses quant à la modélisation des tâches d'insertions simples. Le guidage sur géométries abstraites, combiné à l'activation de contraintes cinématiques (pivot et glissière) a montré de réels gains en performance et en compréhension de ce type d'assem-

blage. L'exemple industriel utilisant un double guidage de ce type nous conforte dans l'idée que la méthode VCG possède un réel potentiel. Il est toutefois nécessaire d'évaluer d'autres modélisations d'assemblage et d'autres formes de guidage géométrique. Il nous paraît essentiel d'évaluer, comme nous l'avons fait pour le *peg-in-hole*, l'activation des liaisons : appui plan bilatéral, hélicoïdale et linéaire annulaire avec différentes modélisation des géométries guides (abstraites).

Ces travaux vont faire l'objet d'une implémentation opérationnelle dans les produits de la société Haption, et notamment dans le module IPP (*interactive physic pack*) de CATIA. En travaillant directement dans le modeleur CAO, nous profiterons de l'échange direct des données géométriques et des données d'assemblage pour mettre en place les VCG.

Bibliographie

- [ABD⁺99] V. Acary, J. Y. Blaide, P. Drap, M. Florenzano, S. Garrec, M. Jean, and D. Merad. NSCD method applied to mechanical simulation of masonry in historical buildings using moma. *XVII CIPA International Symposium. International Committee for Architectural Photogrammetry*), 1999.
- [AGJ05] F. Amirabdollahian, G. T. Gomes, and G. R. Johnson. The peg-in-hole : a VR-based haptic assessment for quantifying upper limb performance and skills. In *IEEE International Conference on Rehabilitation Robotics*, pages 422–425, 2005.
- [AH98] R. J. Adams and B. Hannaford. A two-port framework for the design of unconditionally stable haptic interfaces. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, pages 1254–1259, 1998.
- [AH99] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. In *Proceedings of the IEEE Transaction on robotics and automation*, volume 15, 1999.
- [Ani03] M. Anitescu. A fixed time-step approach for multibody dynamics with contact and friction. In *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, volume 4, pages 3725–3731, 2003.
- [APS99] M. Anitescu, F. A. Potra, and D. E. Stewart. Time-stepping for three dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177 :183–197, 1999.
- [Bar89] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proceedings of SIGGRAPH '89*, pages 223–232. ACM Press, 1989.
- [Bar91] D. Baraff. Coping with friction for non-penetrating rigid body simulation. In *Proceedings of SIGGRAPH '91*, pages 31–40. ACM Press, 1991.

- [Bar94] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of SIGGRAPH '94*, pages 23–34. ACM Press, 1994.
- [BB88] R. Barzel and A. H. Barr. A modeling system based on dynamic constraints. In *Proceedings of SIGGRAPH '88*, pages 179–188, 1988.
- [BCP⁺08] P. Bourdot, T. Convard, F. Picon, M. Ammi, D. Touraine, and J. M. Vézien. VR-CAD integration : multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. *Computer-Aided Design*, 2008.
- [BDP⁺02] B. Brogliato, A. A. Ten Dam, L. Paoli, F. Genot, and M. Abadie. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *Applied Mechanics Reviews*, 55(2) :107–150, 2002.
- [BHS95] P. J. Berkelman, R. L. Hollis, and S. E. Salcudean. Interacting with virtual environments using a magnetic levitation haptic interface. In *Proceedings of the 1995 International Conference on Intelligent Robots and Systems*, pages 117–122, 1995.
- [BJ07] J. Barbič and D. L. James. Time-critical distributed contact for 6-DOF haptic rendering of adaptively sampled reduced deformable models. In *2007 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 171–180, 2007.
- [BML⁺04] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G.D. Hager. Vision-assisted control for manipulation using virtual fixtures. *IEEE Transactions on Robotics*, 20(6) :953–966, 2004.
- [BS96] G. Butlin and C. Stops. CAD data repair. In *Proceedings of the 5th International Meshing Roundtable*, pages 7–12, 1996.
- [Bur99] G. C. Burdea. Haptic feedback for virtual reality. In *7th International Scientific Workshop on Virtual Reality and Prototyping*, June 1999.
- [CB94] J. E. Colgate and J. M. Brown. Factors affecting the Z-width of a haptic display. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3205–3210, 1994.
- [CK86] R. K. Culley and K. G. Kempf. A collision detection algorithm based on velocity and distance bounds. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1064–1069, 1986.
- [CLMP95] J. D. Cohen, M. C. Lin, D. Manocha, and K. Ponamgi. I-COLLIDE : an interactive and exact collision detection system for large-scale

- environments. In *Proceedings of ACM Int. 3D Graphics Conference*, pages 189–196, 1995.
- [CPS92] R. W. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, February 1992.
- [CSB95] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *Proceedings of the IEEE International Conference on Human Robot Interaction and Cooperative Robots*, volume 3, pages 140–145, 1995.
- [CSC05] D. Constantinescu, S. E. Salcudean, and E. A. Croft. Haptic rendering of rigid contacts using impulsive and penalty forces. *IEEE Transaction on Robotics*, 21(3) :309–323, 2005.
- [DAK04] C. Duriez, C. Andriot, and A. Kheddar. Signorini’s contact model for deformable objects in haptic simulations. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pages 3232–3237, 2004.
- [DCRS97] R. G. Dewar, I. D. Carpenter, J. M. Ritchie, and J. E. L. Simmons. Assembly planning in a virtual environment. In *Conference on management and technology*, pages 664–667, 1997.
- [DDKA06] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. In *Proceedings of IEEE Transaction on Visualization and Computer Graphics*, volume 12, pages 36–47, 2006.
- [Dum05] G. Dumont. *Simulation mécanique et réalité virtuelle : outils d’aide à la compréhension et à la conception des systèmes*. HDR - Habilitation à diriger des recherches, Université de Rennes 1, Novembre 2005.
- [EB02] M. O. Ernst and M. S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature, Letter to Nature*, 415 :429–433, 2002.
- [EL00] S. A. Ehmann and M. C. Lin. Accelerated proximity queries between convex polyhedra by multi-level Voronoi marching. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.
- [FL01] S. Fischer and M. C. Lin. Fast penetration depth estimation for elastic bodies using deformed distance fields. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 330–336, 2001.

- [FMT⁺03] P. Fuchs, G. Moreau, J. Tisseau, P. Guiton, B. Arnaldi, A. Berthoz, J. M. Burkhardt, and S. Coquillart. *Le traité de la réalité virtuelle, 3ème édition*, volume 3. Presses de l'Ecole des Mines de Paris, 2003.
- [GC97] R. B. Gillespie and J. E. Colgate. A survey of multibody dynamics for virtual environments. In *Proceedings of the ASME Symposium on Dynamic Systems and Control Division*, pages 45–54, 1997.
- [GJK88] E. G. Gilbert, D. W. Jonhson, and S. Keerty. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4 :193–203, 1988.
- [GLM96] M. Gottschalk, M. C. Lin, and D. Manocha. OBBTree : a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH 96*, volume ACM Press, pages 171–180, 1996.
- [Glo99] C. Glocker. Formulation of spatial contact situations in rigid multibody systems. *Computer Methods in Applied Mechanics and Engineering*, 177(3) :199–214, July 1999.
- [Goe64] R. C. Goertz. Manipulator systems development at ANL. In *Proceedings of the 12th Conference on Remote Systems Technology, American Nuclear Society*, pages 117–136, 1964.
- [Gos05] F. Gosselin. Optimisation des interfaces haptiques : problèmes, méthodes, applications. In *17ème Congrès Français de Mécanique*, 2005.
- [HACH⁺04] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24 :16–29, 2004.
- [HFKS03] S. Hasegawa, N. Fujii, Y. Koike, and M. Sato. Real-time rigid body simulation based on volumetric penalty method. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 326–332, 2003.
- [HHJS99] S. Hartmann, M. Hiller, A. Jennert, and M. Sabinarz. CAD-based contact modelling for multibody systems. *IUTAM Symposium on Unilateral Multibody Contacts, Solid Mechanics And Its Applications*, 1999.
- [HLC⁺97] T. C. Hudson, M. C. Lin, J. Cohen, S. Gottschalk, and Dinesh Manocha. V-COLLIDE : accelerated collision detection for VRML. In *VRML 97 : Second Symposium on the Virtual Reality Modeling Language*, 1997.

- [Hog88] N. Hogan. On the stability of manipulators performing contact tasks. In *Proceeding of the IEEE Transaction on Robotics and Automation*, volume 4, pages 677–686, 1988.
- [HS04] S. Hasegawa and M. Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. In *EUROGRAPHICS 2004*, volume 23, 2004.
- [HWN04] J. D. Hwang, M. D. Williams, and G. Niemeyer. Toward event-based haptics : rendering contact using open-loop force pulses. In *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 24–31, 2004.
- [IML07] R. Iacob, P. Mitrouchev, and J. C. Léon. Modélisation des trajectoires d’assemblage et de désassemblage associées aux liaisons fonctionnelles de produits. In *Actes du 18ème Congrès Français de Mécanique 18-ème Congrès Français de Mécanique*, 2007.
- [IML08] R. Iacob, P. Mitrouchev, and J. C. Leon. Contact identification for assembly-disassembly simulation with a haptic device. *The Visual Computer*, 24(11) :973–979, 2008.
- [JC01] D. E. Johnson and E. Cohen. Spatialized normal cone hierarchies. In *Proceedings of ACM SIGGRAPH Symp. Interactive 3D Graphics*, pages 129–134, March 2001.
- [Jea99] M. Jean. The non-smooth-contact-dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177 :235–257, 1999.
- [Jou06] L. Joussemet. *Approche évolutionniste pour la détection des collisions au sein d’environnements virtuels denses*. PhD thesis, Université de Montpellier 2, 2006.
- [JP01] D. L. James and D. K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e*, 2(1), 2001.
- [JP04] D. L. James and D. K. Pai. BD-Tree : output-sensitive collision detection for reduced deformable models. In *ACM Transactions on Graphics - SIGGRAPH '04*, volume 23, pages 393–398, 2004.
- [JPS93] H.H. Jo, H. R. Parsaei, and W. G. Sullivan. *Concurrent engineering : contemporary issues and modern design tools*. Chapman & Hall, 1993.
- [JTT01] P. Jimenez, F. Thomas, and C. Torras. 3D collision detection : a survey. *Computers and Graphics*, 25 :269–285, 2001.
- [JW03] D. E. Johnson and P. Willemsen. Six degree-of-freedom haptic rendering of complex polygonal models. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, page 229, 2003.

- [JW04] D. E. Johnson and P. Willemsen. Accelerated haptic rendering of polygonal models through local descent. In *Proceedings of the 12th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 18–23, 2004.
- [JWC05] D. E. Johnson, P. Willemsen, and E. Cohen. Six degree-of-freedom haptic rendering using spatialized normal cone search. In *Proceedings of the IEEE Transactions on Visualization and Computer Graphics*, volume 11, pages 661–670, 2005.
- [JWJ⁺99] S. Jayaram, Y. Wang, U. Jayaram, H. Tirumali, K. Lyons, and P. Hart. VADE : a virtual assembly design environment. *IEEE Computer Graphics and Applications*, 19 :44–50, 1999.
- [KEP05] D. M. Kaufman, T. Edmunds, and D. K. Pai. Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics SIGGRAPH 2005*, 24(3) :946–956, 2005.
- [KFN05] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. Event-based haptics and acceleration matching : portraying and assessing the realism of contact. In *Proceedings of the IEEE First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 381–387, 2005.
- [KFN06] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. In *Proceedings of the IEEE Transactions on Visualization and Computer Graphics*, volume 12, pages 219–230, March-April 2006.
- [Khe97] A. Kheddar. *Téléopération basée sur le concept du robot caché*. PhD thesis, Université Pierre et Marie Curie, 1997.
- [KHI⁺07] S. Kockara, T. Halic, K. Igbal, C. Bayrak, and R. Rowe. Collision detection : a survey. In *Proceedings of the IEEE International conference on Systems Man and Cybernetics*, pages 4046–4051, 2007.
- [KHKS02] S. Kim, S. Hasegawa, Y. Koike, and M. Sato. Tension based 7-DOF force feedback device : SPIDAR-G. In *Proceedings of the IEEE Virtual Reality Conference*, pages 283–284, 2002.
- [KNF04] S. Kimmerle, M. Nesme, and F. Faure. Hierarchy accelerated stochastic collision detection. *Publication Evasion - Laboratoire Jean Kuntzmann - INRIA Rhône-Alpes*, 2004.
- [KPZ⁺04] A. B. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. L. MacKenzie. Assembling virtual fixtures for guidance in training environments. *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on*, pages 367–374, 2004.

- [Kru99] M. Krus. *Connexions et Facettisation : gestion adaptative de scènes virtuelles*. PhD thesis, Université de Paris 11, 1999.
- [KSZB95] H. Keller, H. Stolz, A. Ziegler, and T. Braunl. Virtual mechanics simulation and animation of rigid body systems : Aero. *Simulation*, 65(1) :74–79, 1995.
- [LAC03] A. Lecuyer, C. Andriot, and A. Crosnier. Interfaces haptiques et pseudo-haptiques. In *Proceedings of JNRR*, pages 43–48, 2003.
- [Law93] D. A. Lawrence. Stability and transparency in bilateral teleoperation. In *Proceedings of the IEEE Transactions On Robotics And Automation*, volume 9, 1993.
- [LC91] M. C. Lin and J. F. Canny. A fast algorithm for incremental distance calculation. In *Proceedings of the IEEE Transactions on Robotics and Automation*, volume 2, pages 1008–1014, 1991.
- [LCN99] J. C. Lombardo, M. P. Cani, and F. Neyret. Real-time collision detection for virtual surgery. *Computer Animation '99*, 1999.
- [LG98] M. C. Lin and S. Gottschalk. Collision detection between geometric models : a survey. In *Proceeding of IMA Conference on Mathematics of Surfaces*, pages 37–56, 1998.
- [LGLM99] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. Technical report, Technical report TR99-018, Department of Computer Science, University of N. Carolina, 1999.
- [LIT07] M. Li, M. Ishii, and R. H. Taylor. Spatial motion constraints using virtual fixtures generated by anatomy. *IEEE Transactions on Robotics*, 23(1) :4–19, 2007.
- [LMCL01] S. Lertpolpairoj, T. Maneewarn, S. Charoenseang, and D. Laowattana. The effect of static virtual couplings on realistic performance of haptic systems. *Asian Conference on Robotics and its Applications*, 2001.
- [LML02] S. Lertpolpairoj, T. Maneewarn, and D. Laowattana. Realistic and stability performances of haptic system with adaptive virtual coupling. In *Proceedings of the IEEE Transactions on Robotics and Automation*, 2002.
- [LRD⁺07] T. Lim, J. Ritchie, R. Dewar, J. Corney, P. Wilkinson, M. Calis, M. Desmulliez, and J. Fang. Factors affecting user performance in haptic assembly. *Virtual Reality*, 11(4) :241–252, 2007.
- [LW05] T. Liu and M. Y. Wang. Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping

- and Gauss-Seidel methods. In *Proceedings of the IEEE Transactions on Automation Science and Engineering*, volume 2, pages 19–31, 2005.
- [LX04] Q. Luo and J. Xiao. Physically accurate haptic rendering with dynamic effects. In *Proceedings of the IEEE on Computer Graphics and Animation*, volume 24, pages 60–69, 2004.
- [Löt81] P. Lötstedt. Coulomb friction in two dimensional rigid body systems. In *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, pages 605–615, 1981.
- [Löt82] P. Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal on Applied Mathematics*, 42(2) :281–296, 1982.
- [Mab98] M. Mabrouk. A unified variational model for the dynamics of perfect unilateral constraints. *European Journal of Mechanics - A/Solids*, 17(5) :819–842, 1998.
- [MC95] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Proceedings of Symposium on Interactive 3D Graphics*, 1995.
- [MCF00] B. E. Miller, J. E. Colgate, and R. A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. In *Proceedings of the IEEE Transaction on Robotics and Automation*, volume 6, 2000.
- [MH03] M. Mahvash and V. Hayward. Passivity-based high-fidelity haptic rendering of contact. In *Proceedings of the IEEE International Conference on Intelligent Robotics and Automation*, volume 3, pages 3722–3728, 2003.
- [MH05] M. Mahvash and V. Hayward. High-fidelity passive force-reflecting virtual environments. In *Proceedings of the IEEE International Conference on Intelligent Robotics and Automation*, pages 38–46, 2005.
- [Mik04] A. Mikchevitch. *Méthodologie de simulation d'assemblage/désassemblage de composants flexibles : application à la simulation de pièces de type poutres flexibles dans un environnement de réalité virtuelle*. PhD thesis, Institut National Polytechnique de Grenoble, 2004.
- [Mir00] B. Mirtich. Timewarp rigid body simulation. In *Proceedings of SIGGRAPH '00*, pages 193–200, 2000.
- [MKF03] P. Meseure, A. Kheddar, and F. Faure. Détection de collision et calcul de la réponse. *CNRS - Action Spécifique No 90*, 2003.

- [Mor88] J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. *Non smooth mechanics and applications, CISM Courses and Lectures*, 302 :1–82, 1988.
- [Mor94] J. J. Moreau. Some numerical methods in multibody dynamics : application to granular materials. *European journal of mechanics. A. Solids*, 13 :93–114, 1994.
- [MPT99] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of SIGGRAPH'99*, pages 401–408, 1999.
- [MPT06] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Voxel-based 6-DOF haptic rendering improvements. *Haptics-e*, 3(7), 2006.
- [MPT07] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. A robust, efficient and time-stepping compatible collision detection method for non-smooth contact between rigid bodies of arbitrary shape. In *MULTIBODY DYNAMICS, ECCOMAS Thematic Conference*, 2007.
- [MS94] T. H. Massie and J. K. Salisbury. The PHANTOM haptic interface : a device for probing virtual objects. In *Proceedings of the ASME Symposium on Haptic Interfaces for Virtual Environment*, 1994.
- [MW88] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Proceedings of SIGGRAPH '88*, volume ACM Press, pages 289–298. ACM, 1988.
- [MYS⁺90] M. Minsky, M. Ouh Young, O. Steele, F. P. Brooks, and M. Behensky. Feeling and seeing issues in force display. In *Proceedings of SIGGRAPH '90*, volume 24, pages 235–243, 1990.
- [MZ90] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. In *Proceedings of SIGGRAPH '90*, volume 24, pages 29–38, 1990.
- [NS97] G. Niemeyer and J. J. E. Slotine. Designing force reflecting teleoperators with large time delays to appear as virtual tools. In *1997 Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2212–2218, 1997.
- [OL06] M. A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. In *Proceedings of IEEE Transactions on Robotics*, volume 22, pages 94–101, 2006.
- [ORC07] M. Ortega, S. Redon, and S. Coquillard. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. In *Proceedings of the IEEE Transactions on Visualization and Computer Graphics*, volume 13, pages 458–469, 2007.

- [Ota00] S. Otamane. *Télétravail robotisé et réalité augmentée : application à la téléopération via internet*. PhD thesis, Université d'Evry-Val d'Essonne, 2000.
- [PML95] M. K. Ponamgi, D. Manocha, and M. C. Lin. Incremental algorithms for collision detection between solid models. In *Proceedings of the third ACM Symposium on Solid Modelling and Applications*, pages 293–304, 1995.
- [PPG04] M. Pauly, D. K. Pai, and L. J. Guibas. Quasi-rigid objects in contact. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2004.
- [PS02] S. Payandeh and Z. Stanisic. On application of virtual fixtures as an aid for telemanipulation and training. *Haptic Interfaces for Virtual Environment and Teleoperator Systems*, page 18, 2002.
- [PvdDJ⁺01] D. K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lang, J. L. Lloyd, J. L. Richmond, and S. H. Yau. Scanning physical interaction behavior of 3D objects. In *Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, pages 87–96, 2001.
- [RA06] M. Renouf and V. Acary. Comparison and coupling of algorithms for collisions, contact and friction in rigid multi-body simulations. In *European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering*, 2006.
- [RAD05] M. Renouf, V. Acary, and G. Dumont. 3D frictional contact and impact multibody dynamics : a comparison of algorithms suitable for real-time applications. In *ECCOMAS Thematic Conference Multi-body Dynamics 2005*. ECCOMAS, 21-24 june 2005.
- [RDA04] M. Renouf, F. Dubois, and P. Alart. A parallel version of the non smooth contact dynamics algorithm applied to the simulation of granular media. *Journal of Computational and Applied Mathematics - Selected Papers from the Second International Conference on Advanced Computational Methods in Engineering*, 168(1-2) :375–382, 2004.
- [Ren04] M. Renouf. *Optimisation numérique et calcul parallèle pour l'étude des milieux divisés bi et tridimensionnels*. PhD thesis, Université de Montpellier 2, 2004.
- [RH94] C. Ramstein and V. Hayward. The PANTOGRAPH : a large workspace haptic device for a multi-modal human-computer interaction. *Conference on Human Factors in Computing Systems*, 1994.

- [RKC00] S. Redon, A. Kheddar, and S. Coquillart. An algebraic solution to the problem of collision detection for rigid polyhedral objects. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 3733–3738, 2000.
- [RKC01] S. Redon, A. Kheddar, and S. Coquillart. CONTACT : arbitrary in-between motions for collision detection. In *Proceedings of IEEE International Workshop on Robot-Human Interactive Communication*, 2001.
- [RKK97] D. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proceedings of the 24th annual conference on Computer Graphics and Interactive Techniques*, pages 345–352, 1997.
- [Ros92] L. B. Rosenberg. The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments. *Technical Report, No. AL-TR-1992-XXX*, 1992.
- [Ros93] L. B. Rosenberg. Virtual fixtures : perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium*, pages 76–82, 1993.
- [RPP⁺01] M. Renz, C. Preusche, M. Potke, H. P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted Voxmap-PointShell algorithm. In *Proceedings of the Eurohaptics Conference*, 2001.
- [Sau04] G. Saussine. *Contribution à la modélisation de granulats tridimensionnels : application au ballast*. PhD thesis, Université de Montpellier 2, 2004.
- [SCB04] J. K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering : introductory concepts. In *Proceedings of the IEEE Computer Graphics and Applications*, volume 24, pages 24–32, 2004.
- [SCM04] C. Le Saux, F. Cevaer, and R. Motro. Contribution to 3D impact problems : collision between two slender bars. *Comptes rendus de l'Académie des Sciences : Mécanique*, 332 :17–22, 2004.
- [Set07] A. Seth. *Combining physical constraints with geometric constraint-based modeling for virtual assembly*. PhD thesis, Iowa State University, 2007.
- [SS98] J. Sauer and E. Schomer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proceedings of the ACM symposium on Virtual Reality Software and Technology*, pages 153–162, 1998.

- [ST96] D. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39 :2673–2691, 1996.
- [Ste98] D. Stewart. Convergence of a time-stepping scheme for rigid body dynamics and resolution of Painlevé’s problem summary. In *Archive for Rational Mechanics and Analysis*, volume 145, pages 215–260, 1998.
- [SV97] S. E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. *ASME Journal of Dynamical Systems*, 119 :127–132, 1997.
- [SVO07] A. Seth, J. Vance, and J. H. Oliver. Combining geometric constraints with physics modeling for virtual assembly using SHARP. In *ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, 2007.
- [TD08] L. Tching and G. Dumont. Interactive simulation based on non-smooth contact dynamics : application to haptic rigid-body simulations. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, pages 87–90, 2008.
- [TDAP09] L. Tching, G. Dumont, B. Arnaldi, and J. Perret. Interactions haptiques au sein de simulations dynamiques : traitement dynamique des contacts et des chocs entre objets rigides. *RSTI - TSI*, 28(8) :953–981, 2009.
- [TKH⁺04] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F., N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. In *Proceedings of Eurographics*, pages 119–139, 2004.
- [TKI99] J. Takamatsu, H. Kimura, and K. Ikeuchi. Classifying contact states for recognizing human assembly tasks. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, volume 4, pages 177–182, 1999.
- [UNT⁺02] B. J. Unger, A. Nicolaidis, A. Thompson, R. L. Klatzky, R. L. Hollis, P. J. Berkelman, and S. Lederman. Virtual peg-in-hole performance using a 6-DOF magnetic levitation haptic device : comparison with real forces and with visual guidance alone. In *HAPTICS ’02 : Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 263–270. IEEE Computer Society, 2002.

- [vdB98] G. van der Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2 :1–13, 1998.
- [vdB99] G. van der Bergen. A fast and robust GJK implementation for collision detection of convex objects. *Journal of Graphics Tools*, 4(2) :7–25, 1999.
- [vGS01] C. van Geem and T. Simeon. KCD : a collision detector for path planning in factory models. *LAAS Rep. 2001-073*, 2001.
- [Yua02] X. Yuan. Interactive assembly planning in virtual environments. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1462–1467, 2002.
- [ZRLK07] X. Zhang, S. Redon, M. Lee, and Y. J. Kim. Continuous collision detection for articulated models using Taylor models and temporal culling. In *Proceedings of SIGGRAPH '07*, volume 26, 2007.
- [ZS95] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 146–151, 1995.

Table des figures

1.1	Interaction haptique	14
1.2	Schéma de principe d'un système haptique local	15
1.3	Différentes combinaisons de couplage entre le périphérique et le simulateur.	17
1.4	Introduction d'un couplage virtuel entre le dispositif haptique et l'objet virtuel du simulateur, lors d'une tâche d'assemblage CAO (courtoisie de Haption).	18
2.1	Schéma classique d'un simulateur physique.	24
2.2	Différentes approches du contact entre solides rigides : par proximité, exacte ou par interpénétration.	25
2.3	De gauche à droite : distance minimale globale - paires en-dessous d'une distance seuil - points de contact - distances minimales locales (d'après (Johnson <i>et al.</i> , 2003)).	30
2.4	Distance de proximité d , entre le PointShell de l'objet candidat et la Voxmap de l'objet antagoniste en contact (d'après (McNeely <i>et al.</i> , 2006)).	31
2.5	Contact entre deux solides A et B, en un point α , dont l'effort de contact est décomposable en une composante normale n et une composante tangentielle t	32
2.6	Visualisation des données sous OpenMask (à gauche) et Ogre (à droite).	39
2.7	Architecture de Confetis.	40
3.1	Représentation 3D du changement du nombre de DDL entre objets, entre les géométries théoriques (à gauche) et les géométries facettisées (à droite).	47
3.2	Processus classique de traitement des objets CAO, pour leur utilisation en simulation de réalité virtuelle	48
3.3	Perte de précision géométrique lors du changement de représentation de la CAO vers la RV, sur un exemple de peg-in-hole	50

3.4	Exemple de localisation de point de contact entre deux roues dentées discrétisées. À gauche le contact est localisé à l'intersection de volumes englobants (boîtes rouges). À droite, le contact est défini comme une distance seuil entre géométries élémentaires.	52
3.5	Dessin technique 2D du montage <i>peg-in-hole</i> , sans et avec changement diminution du diamètre du pion	53
3.6	Conditions de modification des géométries du pion, pour l'utilisation dans un algorithme de détection de type <i>LMD</i>	54
3.7	Exemple <i>peg-in-hole</i> où le diamètre du pion est non ajusté (à droite) et ajusté (à gauche), au sein de la simulation de RV. . . .	55
3.8	Décomposition du guidage. À gauche, l'objet est en phase exploratoire, aucun guidage n'est activé. Au milieu, l'objet est en phase d'approche, guidé par un guidage géométrique (non représenté). À droite, l'objet est en phase d'assemblage, guidé par un guidage contraint (non représenté).	59
3.9	Vue 3D d'une liaison nulle entre la pièce grise et la pièce verte . .	62
3.10	Vue 3D d'une liaison encastrement entre la pièce grise et la pièce verte	63
3.11	Vue 3D d'une liaison ponctuelle bilatérale entre la pièce grise et la pièce verte	64
3.12	Vue 3D d'une liaison linéaire rectiligne entre la pièce grise et la pièce verte	65
3.13	Vue 3D d'une liaison appui plan bilatérale entre la pièce grise et la pièce verte	66
3.14	Vue 3D d'une liaison glissière entre la pièce grise et la pièce verte	67
3.15	Vue 3D d'une liaison pivot entre la pièce grise et la pièce verte . .	68
3.16	Vue 3D d'une liaison pivot-glissant entre la pièce grise et la pièce verte	69
3.17	Vue 3D d'une liaison hélicoïdale entre la pièce grise et la pièce verte	69
3.18	Vue 3D d'une liaison rotule entre la pièce grise et la pièce verte .	70
3.19	Vue 3D d'une liaison linéaire annulaire entre la pièce grise et la pièce verte	71
3.20	Création des guides virtuels, au sein du simulateur, à partir des géométries maillées.	75
3.21	Guidage géométrique entre un pion cylindrique et une surface conique.	77
3.22	Guidage géométrique entre un prisme en "T" et une surface complexe.	78

3.23	Positionnement requis pour activer la contrainte mécanique. Sur la figure de gauche, le positionnement d'un pion cylindrique est donné par la coïncidence de leurs axes. À gauche, le positionnement d'un pion prismatique (base carrée) est donnée par la coïncidence de trois arêtes.	79
3.24	Exemple de guides géométriques utilisés pour le positionnement d'un pion au dessus d'un trou.	79
3.25	Illustration du guidage géométrique, vue en 2D, pour l'assemblage <i>peg-in-hole</i>	80
3.26	Guide géométrique rotatif pour l'application <i>peg-in-hole</i>	81
3.27	Les trois étapes du guidage virtuel contraint, vu de manière schématique (ligne du haut) et vu de manière pratique (ligne du bas)	83
3.28	Grafcet de fonctionnement normal, point de vue utilisateur	85
3.29	Grafcet de fonctionnement normal, point de vue simulateur. . . .	87
4.1	Manipulation effectuée sur un bras Virtuose 6D-35/45.	94
4.2	Interaction entre la sphère (verte) contrôlée par l'utilisateur et 50 autres sphères soumises à la gravité.	95
4.3	Interaction entre la sphère (verte) contrôlée par l'utilisateur un mur de 22 briques (jaunes).	96
4.4	Estimation des performances temporelles de Confetis, en fonction du nombre de sphères et du nombre moyen de contacts simultanés	96
4.5	Interactions, en salle immersive, entre un utilisateur et une ligne de dominos.	98
4.6	Assemblage d'objets CAO avec un périphérique haptique Virtuose.	101
4.7	Le guidage virtuel contraint (VCG) appliqué, en situation, au <i>peg-in-hole</i> . De gauche à droite : le pion vert est positionné à la verticale au dessus du trou à l'aide du guidage géométrique, puis les collisions sont désactivées et la contrainte cinématique est appliquée (le pion devient bleu).	104
4.8	Deux différentes façons de positionner le guide fixe au dessus des blocs.	106
4.9	Différentes combinaisons de mouvement entre les 4 trous.	107
4.10	Les mesures temporelles sont réalisées entre le moment où le pion approche de la zone d'assemblage et le moment où le pion est au fond du trou.	109
4.11	Démonstration préliminaire.	110
4.12	Amélioration des performances au cours des expériences, due à l'effet d'apprentissage.	114
4.13	Convergence de l'écart-type, due à l'effet d'apprentissage.	114

4.14	Moyennes des temps d'insertion, pour l'ensemble des sujets, classées par cibles et cas d'assemblage.	115
4.15	Moyennes des écarts-types, pour l'ensemble des sujets, classées par cibles et cas d'assemblage.	115
4.16	Répartition des performances temporelles, pour les cas 5 à 8 . . .	117
4.17	Estimation (subjective) du niveau de manipulation haptique . . .	118
4.18	Le cas 8, qui propose l'interaction haptique et la visualisation des deux guides, offre les meilleures performances temporelles.	118
4.19	Estimation (subjective) du niveau de manipulation haptique. . . .	119
4.20	Moyenne des performances, en fonction des trajectoires du pion. .	123
4.21	Traitement des données du questionnaire	124
4.22	Montage d'un moteur d'essuie-glace : prétraitement des données 3D. Courtoisie de Renault.	125
4.23	Implementation de la méthode VCG sur un cas industriel : l'assemblage d'un moteur d'essuie-glace (objet vert) dans le châssis d'une automobile.	126

Résumé

La simulation d'assemblage CAO en environnement virtuel donne une nouvelle dimension au développement rapide et à moindre coût d'un produit, en permettant de travailler collaborativement autour de la maquette numérique. L'interaction entre l'utilisateur et la simulation suppose que les modèles mécaniques utilisés soient conformes à la réalité, dans la mesure où les résultats de la simulation doivent être proches du comportement des objets matériels réels. Dans cette thèse, nous nous sommes intéressés au traitement des interactions entre objets 3D, dans le cadre spécifique des assemblages haptiques. L'objectif de la thèse est d'étudier les différents modes de dynamique applicables dans les simulations d'assemblage CAO. Ce domaine de recherche comporte plusieurs verrous technologiques et scientifiques concernant le couplage des interfaces haptiques avec des moteurs de simulation dynamique. La contrainte de temps interactif, très restrictive, impose à toutes méthodes dynamiques de détecter des collisions et calculer des efforts de contact de manière très rapide et avec des critères forts de stabilité mécanique.

Dans ce contexte, nous proposons d'appliquer les technologies de l'interaction haptique, pour la réalisation d'assemblages mécaniques issus de la CAO. Parmi les nombreux facteurs qui influencent le degré de réalisme d'une simulation interactive d'assemblage, nous nous concentrons sur le réalisme des comportements physiques des objets 3D simulés ainsi que sur les performances de l'utilisateur dans la réalisation de tâches d'assemblage. Dans le cadre précis des simulations haptiques, nous proposons de traiter simultanément ces deux facteurs. Au sein d'une même simulation d'assemblage, l'utilisateur doit pouvoir percevoir des comportements dynamiques réalistes lors d'interactions élémentaires avec l'environnement virtuel, ainsi que pouvoir réaliser des tâches d'A/D de manière efficace.

Dans les simulations d'assemblage, nous avons identifié deux phases de manipulation où le traitement dynamique des contacts peut être réalisé : soit de manière contrainte pour aider les utilisateurs à positionner des objets dans l'espace, soit de manière non contrainte pour obtenir un rendu haptique réaliste des interactions. L'approche proposée consiste donc à séparer les interactions liées à l'exploration des scènes haptique, des interactions liées aux assemblages. Au sein d'une même simulation d'assemblage, nous proposons d'utiliser des modèles de traitement dynamique non contraint pour les phases exploratoires et l'utilisation de modèles de dynamique contrainte pour réaliser les assemblages haptiquement. Dans une première contribution, nous proposons de baser la simulation haptique sur une modélisation rigoureuse de la mécanique : pour rendre compte de comportements réalistes des objets, nous utilisons les méthodes de dynamique non régulière, et particulièrement des approches de non smooth contact dynamics (NSCD). Dans une deuxième contribution, nous proposons d'identifier les différentes étapes

qui composent l'assemblage interactif, pour assister l'utilisateur dans la réalisation de tâches de montage et de démontage. Cette assistance consiste en un guidage virtuel des objets, qui contraint géométriquement et cinématiquement les mouvements des objets à assembler. Nous présentons une nouvelle méthode de guidage, baptisée guidage virtuel contraint, qui applique des contraintes cinématiques aux objets, en fonction des tâches à réaliser. La transition entre le mode exploratoire, géré par dynamique non contrainte et le mode assemblage, géré par dynamique contrainte, est assuré par un guidage géométrique inspiré des virtual fixtures, qui constitue une troisième contribution.

Abstract

In this thesis, we are interested in the treatment of interactions between 3D objects in the specific context of haptic assemblies. The aim of the thesis is to study the different dynamic modes that are applicable to the simulation of CAD assemblies. This field of research involves several scientific and technologic issues on the coupling of haptic interfaces with dynamic simulation engines. The interactive-time constraint forces dynamic engine to detect collisions and compute the contact forces very quickly, following a stability criterion.

In this context, we propose to apply the technology of haptic interaction to manage mechanical assemblies of CAD objects. Among the many factors that influence the degree of realism of an interactive simulation of assembly, we focus on the realism of the behaviour of physical objects and on the performance of users achieving assembly tasks. In the context of haptic simulations, we propose to address these two factors simultaneously. Within a single assembly simulation, the user will feel realistic dynamic behaviour during elementary interactions with the virtual environment, while performing A/D tasks in an efficient way.

In the simulations of assembly, we identified two handling phases where the contact dynamics can be performed : either in a constrained way to help users to position objects in space, or in a non-constrained way to return realistic haptic interactions. The approach is to separate the interactions related to the exploration scenes, from the interactions related to assemblies. Within a single simulation, we propose to use unconstrained dynamic models to process exploratory phase and the use of constrained dynamic models to process haptic assembly. In a first contribution, we propose to use a rigorous modelling of mechanics to simulate realistic behaviours of objects : we use the methods of non-smooth dynamics and especially the non-smooth contact dynamics (NSCD) approaches. In a second contribution, we propose to identify the different stages that compose the interactive assembly to assist the user. This assistance consists in a virtual guidance composed of geometrical and kinematical constraints applied to the control of

the objects to assemble. We present a new method of guidance, called virtual constraint guidance that applies kinematical constraints that depend on the assembly tasks. The transition between the exploratory mode and the assembly mode is made by a geometric guidance inspired by virtual fixtures.